

# **USER'S GUIDE TO THE CG-MATHEW/ADPIC MODELS**

**Version 3.0**

**Authors:**

**Connee Foster, Hoyt Walker, Gordon Duckworth,  
Allan Taylor, Gayle Sugiyama**

**March 1995**

**Lawrence Livermore National Laboratory, University of California  
Regional Atmospheric Sciences Division  
Livermore, California 94550**

#### DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

## **ACKNOWLEDGMENT**

This manual was originally released in 1982 with the following authors: Daniel J. Rodriguez, George D. Greenly, Philip M. Gresho, Rolf Lange, Bryan S. Lawver, Leonard A. Lawson, and Hoyt Walker, with Dan Rodriguez also acting as the editor. Connee Foster assumed the editor/co-author role for subsequent unofficial versions of the manual. Connee was also the editor for the next official version (Rev. 1) of the this document released in December 1992. She was ably assisted in preparing that version by the following people: Robert P. Freis, Kevin T. Foster, John S. Nasstrom, Ronald L. Baskett, Walter W. Schalk III, James S. Ellis, Brenda M. Pobanz, Philip J. Vogt, Leonard A. Lawson, Hoyt Walker, and John C. Pace. Special thanks go to Raylene Cooper for her work in preparing and editing the December 1992 version.

A major re-write by Allan Taylor, Gayle Sugiyama, and Hoyt Walker occurred during Summer 1993, Rev. 2, reflecting the initial implementation of the restructured models. Subsequent informal versions have been edited by Connee Foster with input provided primarily by Allan, Gayle, and Hoyt. Rev. 2 reflects the current state of the models after the restructuring and prior to inclusion of major modernization improvements, such as continuous terrain, conjugate gradient CG-MATHEW, and RDM (random displacement model) and hybrid particles in ADPIC. Special thanks go to Jodi Greenfield for help with scanning figures and editing.

Version Rev 3 includes conjugate gradient CG-MATHEW, generalized map projection, and some other minor changes and additions. Primary authors are Connee Foster, Hoyt Walker, Gordon Duckworth, Allan Taylor, and Gayle Sugiyama.

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

## **PREFACE**

This user's guide documents the CG-MATHEW and ADPIC transport and diffusion models and their supporting stream of codes. It includes a description of each code (Chapter I), its theoretical basis (Chapter II), and the preparation of input and output data (Chapter III).

# Table of Contents

ACKNOWLEDGMENT .....	i
PREFACE .....	ii
<b>I. Introduction to the CG-MATHEW/ADPIC Transport and Diffusion</b>	
<b>Models .....</b>	<b>1</b>
A. CG-MATHEW .....	2
B. ADPIC .....	3
C. CG-MATHEW and ADPIC Support Routines .....	4
1. TOPOG .....	4
2. MEDIC .....	4
3. PLCNT.....	4
4. TIMEHIS.....	5
5. RSUS NOTE	
RSUS is not yet converted (not available).....	5
<b>II. Technical Discussion of ARAC Models .....</b>	<b>6</b>
A. CG-MATHEW .....	6
1. Variational formulation.....	6
2. Boundary Conditions .....	7
3. Derivation of the CG-MATHEW stencil .....	8
4. Conjugate Gradient Algorithm .....	10
5. Output .....	13
B. ADPIC .....	14
1. Formulation—Pseudo-Velocity and particle-in-cell method .....	14
2. Boundary Conditions .....	15
a. Topography.....	15
b. Flux Boundary Conditions.....	15
3. Diffusion Parameters.....	16
a. Horizontal Diffusion Coefficients.....	16
b. Continuous Sources .....	16
c. Instantaneous Clouds.....	17
d. Vertical Diffusivity Coefficient .....	17
4. Options in ADPIC .....	19
a. Source Parameters.....	19
b. Source Geometry .....	19
c. Particle Size Distribution .....	21
d. Dry Deposition .....	21
e. Washout.....	24
f. Plume Rise .....	26
g. Explosive Cloud Rise.....	26
h. Meteorological Parameters.....	28
i. Nested Grid Sampling .....	28
j. Output .....	30
5. Accuracy of ADPIC .....	30
a. Finite Difference Algorithms.....	30

b.	The Number of Eulerian Cells Per Problem .....	31
c.	The Number of Particles Per Cell .....	32
d.	Time-step Calculation .....	33
e.	Sub-grid Scale Diffusion .....	33
C.	Support Routines.....	37
2.	MEDIC .....	40
a.	Parameterized Extrapolation .....	42
b.	Profile Extrapolation .....	46
c.	Terrain Incorporation.....	49
d.	Gridded wind fields .....	49
e.	Hemispheric MEDIC (HMEDIC) .....	50
3.	PLCNT.....	51
4.	TIMEHIS.....	57
5.	RSUS (not implemented yet) .....	58
D.	CG-MATHEW/ADPIC Validation and Error Analysis .....	60
<b>III.</b>	<b>User's Guide .....</b>	<b>63</b>
A.	ARAC Models File System and Flowcharts .....	63
B.	Input Files.....	73
1.	TOPOG.NML .....	74
a.	TOPOG_CONTROL namelist.....	75
b.	TOPOG_PARAMS namelist .....	76
2.	Geography .....	80
3.	Regional Elevation Data.....	80
a.	MEDIC_CONTROL namelist .....	82
b.	MEDIC_PARAMS namelist .....	89
5.	STNLOC.MET .....	99
6.	OBSERV.MET .....	101
7.	CG-MATHEW.NML .....	103
a.	CG-MATHEW_CONTROL namelist .....	103
b.	CG-MATHEW_PARAMS namelist .....	109
8.	MEDIC.VLST.....	114
9.	ADPIC.NML.....	116
a.	ADPIC_CONTROL namelist .....	116
b.	ADPIC_METPARAMS namelist .....	124
c.	ADPIC_SOURCE namelist .....	135
d.	SIGMA_THETA_DATA namelist .....	142
e.	ADPIC_EXPL_CLD namelist .....	143
10.	ADPIC.CLST .....	148
11.	CONC_ANALYSIS.NML.....	148
a.	CONC_ANALYSIS_CONTROL namelist.....	149
b.	PLCNT_GRAPHICS_PARAMS namelist .....	155
c.	PLCNT_LABELING_PARAMS namelist .....	159
d.	TIMEHIS_PARAMS namelist.....	163
12.	TIMEHIS.....	168
13.	RSUSIN.....	170

C. Output Files.....	172
1. TOPOG*.GRID.....	172
2. TOPOG.LOG .....	172
3. TOPOG*.CNML .....	173
4. TOPOG*.CGMB .....	173
5. MEDVEL***.BIF .....	174
6. MEDIC.LOG .....	174
7. MEDIC*.CGMB .....	175
8. MATVEL .....	178
9. CG-MATHEW.VLST .....	179
10. CG-MATHEW.LOG.....	179
11. CG-MATHEW*.CGMB .....	181
12. CONC*.BIF .....	182
13. ADPIC.LOG.....	183
14. ADPIC*.RST .....	185
15. CONC*.RST.....	185
17. ADPIC*.CGMB.....	186
19. CENLINE*.DAT .....	187
20. TIMEHIS.LOG .....	187
APPENDIX A. BIBLIOGRAPHY .....	190
APPENDIX B. TOPOG INTERACTION.....	192
APPENDIX C. TRACEBACK BLOCK DESCRIPTION.....	200
APPENDIX D. ARAC Model System Logical Names .....	206
APPENDIX E. ARAC Model Namelist Summary .....	210

# **I. Introduction to the CG-MATHEW/ADPIC Transport and Diffusion Models**

This user's guide provides both an operational and theoretical summary of the CG-MATHEW/ADPIC transport and diffusion models and supplemental codes needed to execute these models. The acronym, CG-MATHEW, comes from **Mass-Adjusted THrEe-dimensional Wind** field. ADPIC stands for **Atmospheric Diffusion Particle-In-Cell** model. These models calculate air concentration and ground deposition on a regional scale (up to 200 km) from continuous or instantaneous point sources.

CG-MATHEW is a three-dimensional meteorological data-adjustment model which supplies ADPIC with mean wind fields that are mass-consistent and representative of the available meteorological measurements (surface, tower, and upper air soundings). The bottom boundary in this model is determined by grid cell estimates of the actual topographic features of a given site and its environs, which can play a very important role in defining or modeling regional-scale patterns. ADPIC is a three-dimensional, particle-in-cell transport and diffusion code capable of calculating the time-dependent dispersion of inert or radioactive air pollutants.

In addition to CG-MATHEW/ADPIC, three supplemental codes are required to provide input/output support. These codes, which are also discussed in this user's guide, are TOPOG, MEDIC, PLCNT and TIMEHIS. TOPOG defines topographic features for MEDIC, CG-MATHEW, and ADPIC. MEDIC builds a realistic three-dimensional wind field from the observed meteorological conditions. This input is used by CG-MATHEW to compute the mass-consistent wind field required by ADPIC. PLCNT is a post-processing code that allows dose conversion from particle concentration to other units (e.g., Rem). PLCNT uses the resulting doses to generate products for shipment to ARAC site systems and site workstations.

RSUS is another supplemental code which uses the output from PLCNT to construct the input files needed to make a resuspension CG-MATHEW/ADPIC calculation.

An overview of CG-MATHEW/ADPIC and the support routines follows in this section. A complete technical formulation of each code is found in Section II. Section III provides a user's guide to each code.



## I.A. CG-MATHEW

The Conjugate-Gradient Mass-Adjusted THrEe-dimensional Wind (CG-MATHEW) code generates mass-consistent winds by minimal adjustment of input fields derived from observational, analyzed, or model data. The adjustment is performed by a constrained variational minimization using finite-difference methods and a conjugate gradient solution. Generalized symmetric boundary conditions are implemented, which allow control of the relative amounts of normal and tangential adjustment at the lateral and terrain surfaces. Sugiyama et al. (1994) discuss the methodology in detail, give comparisons with potential flow solutions for simple geometries, and demonstrate the use of the new boundary conditions to create different kinds of flow fields, prevent speed reductions, and eliminate flow artifacts caused by complex terrain.

The underlying theoretical framework for the variational adjustment was developed by Sasaki (1958, 1970a, 1970b) and first applied in the Atmospheric Release Advisory Capability (ARAC) operational context by Sherman (1978). A functional is formulated which minimizes the variance between input and output winds subject to a non-divergence constraint, mass-flow boundary conditions on lateral and top faces, and the requirement of zero normal wind-field components at terrain surfaces. The requirement of minimal adjustment maintains consistency with available meteorological measurements (surface and upper air soundings), while atmospheric stability conditions govern the relative amounts of change in the vertical and horizontal wind components. The adjustment is irrotational and neither momentum balance nor energy conservation is imposed. Velocity components are defined on staggered grid faces so that the mass-consistency constraint is cell-flux rather than grid-point based. The problem then reduces to the solution of the Poisson equation in the Lagrange multiplier  $\lambda(x, y, z)$ , with the adjusted winds derived from the corresponding Euler-Lagrange equations.

The original ARAC MATHEW code discretized the Poisson equation using quadratic fits to generate the boundary conditions, which resulted in an asymmetric coefficient matrix. The equation was then solved by a successive overrelaxation (SOR) method. In order to implement a conjugate gradient algorithm which is approximately twice as fast as the SOR approach, the problem is reformulated with standard image point boundary conditions. This yields a symmetric matrix representation consistent with the symmetry of the continuous Poisson problem. The boundary conditions are generalized to include parameters for controlling the relative amounts of normal and tangential adjustment at the lateral and top edges (MIXED boundary condition, Gresho, 1978) and for fixing velocities at the inflow boundaries.

Input wind fields are generally derived by a combination of interpolation and extrapolation from sparse observational data and are typically assumed to be purely horizontal due to the lack of vertical velocity measurements. Such two-dimensional wind fields can be prepared for CG-MATHEW using the MEDIC code which incorporates surface, upper air, and analyzed wind data as available and appropriate. However, the mass-adjustment technique is applicable to input winds with a complete set of  $u$ ,  $v$ , and  $w$  components and can be implemented using three-dimensional wind fields derived from assorted diagnostic and prognostic models.

## **I.B. ADPIC**

ADPIC (Atmospheric Diffusion Particle-in-Cell) is a numerical, three-dimensional, Cartesian, particle-diffusion code capable of calculating the time-dependent distribution of air pollutants under many conditions including strongly distorted wind fields, calm conditions, wet and dry deposition, radioactive decay, and space-and time-variable diffusion parameters.

ADPIC solves the three-dimensional advection-diffusion equation in its flux conservation form (pseudo-velocity technique) for a given mass-consistent advection field (supplied by CG-MATHEW) by finite difference approximations in Cartesian coordinates. The method is based on the particle-in-cell technique with the pollutant concentration represented statistically by Lagrangian marker particles imbedded in an Eulerian grid (Lange, 1973 and 1978). The code is suitable for the study of instantaneous or continuous releases near or on the surface over regional (10 to 200 km) to hemispheric scales.

## **I.C. CG-MATHEW and ADPIC Support Routines**

### **I.C.1. TOPOG**

TOPOG creates the underlying surface for the CG-MATHEW and ADPIC models. The surface is created by averaging terrain data from a regional terrain data base which has, as its source, high resolution elevation data supplied by the Defense Mapping Agency.

TOPOG has three main functions: (1) It verifies that the CG-MATHEW and ADPIC grids, as defined by the user, are oriented correctly with respect to each other and the regional terrain grid. (2) It averages terrain data from the regional data base to the newly defined model grid and then rounds these averaged heights to the nearest model level. This provides the appropriate block topography for the CG-MATHEW and ADPIC models. (3)

### **I.C.2. MEDIC**

MEDIC extrapolates a three-dimensional gridded wind field from a combination of surface and upper air observations. The extrapolation is a complex process, controlled by a number of parameters specified by the user. The extrapolated wind field is sensitive to small changes in some of these parameters and, as a result, widely differing extrapolated fields can be produced, all of which are consistent with a given data set. CG-MATHEW adjusts the winds to achieve mass consistency and then passes them to ADPIC. Because CG-MATHEW makes the smallest necessary adjustments to the extrapolated field and because ADPIC assumes persistence over extended periods (typically one hour), any change in the extrapolated wind field will affect all the results of the ARAC models. Therefore, it is crucial for the user to provide the best extrapolated wind field that can be produced for a data set within the allowed range of the various parameter values. In turn, this means that the graphical output from MEDIC must be examined very carefully by the assessor to determine if the wind field is an acceptable model of the physical state of the atmosphere. If it is not, then MEDIC must be re-executed with revised parameter values until a reasonable representation is produced.

### **I.C.3. PLCNT**

PLCNT uses two-dimensional arrays of nested grid point concentrations (sampling bins) produced by ADPIC as input, passes these concentrations onto another nested grid (a dose bin) using a user-selected dose conversion factor, and combines legitimate combinations of dose bins into another nested grid (a final bin) for subsequent processing. At user-defined times, PLCNT determines an appropriate set of contour levels, and then either draws a set of smoothed contours for shipment to ARAC site systems or writes files with the final bin data for transfer ARAC site workstations. The levels at which contours are drawn can be chosen by the code or specified by the user. If determined by the code, a specified number (up to five) contours will be generated, each differing by an order of magnitude, as a function of the largest grid point concentration. The user has

the option of overriding this choice by initializing a select number of variables in the input file (see Section III.B.8). As an example of the usefulness of this latter option, an environmental standard may have been established for a substance which has been accidentally released into the atmosphere. On entering this value, the code will calculate and plot an isopleth which represents a threshold to this standard, thereby delineating the area of concern.

The model results can be presented in three ways: deposition, integrated air concentration, and instantaneous air concentration. Deposition refers to material deposited on the ground since the beginning of a release. Integration is an accumulation of released material over a designated time and space at a selected level. Instantaneous refers to a picture of a (contoured) concentration field at a moment in time.

PLCNT also has a centerline concentration capability, which starts at a user-specified ADPIC source location and searches along arcs of increasing radius for the maximum along each radial arc. The result series of maxima, along with their locations are available in tabular format, and the locations are also plotted on each Site System contour plot.

#### **I.C.4. TIMEHIS**

In addition to the ability to obtain contours from PLCNT, a companion code TIMEHIS (i.e., Time History) allows concentrations to be estimated using bilinear interpolation for point locations that can be arbitrarily positioned in the grid. The grid point concentrations are developed in exactly the same way as in PLCNT with interpolation to the final bins performed before any smoothing. The times at which the interpolations are made are also determined in the same way as the times of the plots in PLCNT. The series of concentrations are plotted as time series and printed in the TIMEHIS.LOG file. TIMEHIS will also read sampler measurement data, plotting these observations on the time series plots along with the computed values for the same locations and statistics that compare the computed and measured values. This feature has proved useful when comparing model concentrations against collections by actual samplers that have been placed in the field during a tracer experiment.

#### **I.C.5. RSUS NOTE: RSUS is not yet converted (not available)**

RSUS produces a complete set of input files for a CG-MATHEW/ADPIC calculation that will simulate the resuspension, transport, and diffusion of the particles deposited by a previous CG-MATHEW/ADPIC calculation. The resuspension source term is based upon the amount and geometry of the original deposition pattern, as well as other parameters (such as surface characteristics and wind speed) input by the user.

## II. Technical Discussion of ARAC Models

### II.A. CG-MATHEW<sup>+</sup>

#### II.A.1. Variational formulation

A variational problem may be solved by reduction to a set of differential equations known as the Euler equations, which represent necessary, though not sufficient, conditions for a minimum (Courant and Hilbert, 1962). In its simplest form, the CG-MATHEW functional required to minimize the variance between the observed input and adjusted output winds subject to the mass-consistency constraint is

$$J(u, v, w, \lambda) = \int_{\Omega} \frac{1}{2\sigma_H^2} \left( (u - u^o)^2 + (v - v^o)^2 \right) + \frac{1}{2\sigma_v^2} (w - w^o)^2 + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dx dy dz \quad (\text{II.A.1})$$

where  $(u, v, w)$  and  $(u^o, v^o, w^o)$  are the output and input wind components, respectively,  $\lambda(x, y, z)$  is the Lagrange multiplier for the mass-consistency constraint,  $\Omega$  is the volume domain of the simulation, and  $\sigma_v$  and  $\sigma_H$  are parameters controlling the relative amounts of vertical and horizontal adjustment. The Euler-Lagrange equations associated with this functional are

$$\begin{aligned} u - u^o &= \sigma_H^2 \frac{\partial \lambda}{\partial x} \\ v - v^o &= \sigma_H^2 \frac{\partial \lambda}{\partial y} \\ w - w^o &= \sigma_v^2 \frac{\partial \lambda}{\partial z} \end{aligned} \quad (\text{II.A.2})$$

with boundary conditions

$$\begin{aligned} \lambda n_x \delta u|_s &= 0 \\ \lambda n_y \delta v|_s &= 0 \\ \lambda n_z \delta w|_s &= 0 \end{aligned} \quad (\text{II.A.3})$$

Taking appropriate derivatives of the Euler-Lagrange equations and substituting into the mass-consistency constraint,  $\nabla \cdot \vec{u} = 0$ , yields the Poisson equation

$$\sigma_H^2 \left( \frac{\partial^2 \lambda}{\partial x^2} + \frac{\partial^2 \lambda}{\partial y^2} \right) + \sigma_v^2 \frac{\partial^2 \lambda}{\partial z^2} = - \frac{\partial u^o}{\partial x} - \frac{\partial v^o}{\partial y} - \frac{\partial w^o}{\partial z} \quad (\text{II.A.4})$$

which may be solved for  $\lambda$  subject to the boundary conditions. The adjusted wind fields are then determined from Eqs. II.A.2.

---

<sup>+</sup> Abstracted from Sugiyama et al., 1994.

## II.A.2. Boundary Conditions

The boundary conditions of Eqs. II.A.3 can be satisfied if  $\lambda|_S = 0$ . The physical interpretation of this solution is revealed by examination of the Euler-Lagrange equations. If  $\lambda$  is a constant on  $S$ , the derivatives along the surface vanish, which fixes the tangential component of the velocity, while imposing no restrictions on the normal component. This is the ADJUST\_NORMAL condition and is a kind of flow-through boundary. Eqs. II.A.3 are also satisfied if  $\frac{\partial \lambda}{\partial n}|_S = 0$ , which prevents changes in the normal velocity component, while allowing free adjustment of the tangential component. This is a FIXED\_NORMAL or fixed-flow boundary and can be used to maintain zero normal velocities on terrain surfaces <sup>1</sup>.

It is advantageous to incorporate a new surface integral term which provides more flexibility in the boundary conditions and to treat the FIXED\_NORMAL case in a constraint formalism, so that input wind fields with non-zero normal components at terrain surfaces can be used. These considerations lead to a generalized CG-MATHEW functional

$$\begin{aligned} J(u, v, w, \lambda) = & \int_{\Omega} \left[ (\vec{u} - \vec{u}^0) \cdot \mathbf{D}^{-1} \cdot (\vec{u} - \vec{u}^0) + \lambda \nabla \cdot \vec{u} \right] dx dy dz \\ & - \int_{S_{\text{terrain}}} \lambda_{\text{terrain}} \vec{n} \cdot \vec{u} dS \\ & + \frac{1}{2} \int_{S_{\text{flow}}} f(S) (\vec{u} - \vec{u}^0)^2 dS \end{aligned} \quad (\text{II.A.5})$$

where  $f(S)$  is an arbitrary continuous function on the flow-through boundary  $S_{\text{flow}}$ ,  $S_{\text{terrain}}$  is the terrain surface, and  $\mathbf{D}$  is a 3x3 diagonal matrix with non-zero components  $\sigma_i^2 \delta_{i,j}$  ( $\delta_{i,j}$  is the Kronecker delta and  $\sigma_1 = \sigma_2 = \sigma_H$  and  $\sigma_3 = \sigma_V$ ).

The Euler-Lagrange and Poisson equations are unchanged from the simple CG-MATHEW formulation (Eqs. II.A.2 and II.A.4)

$$\vec{u} - \vec{u}^0 = \mathbf{D} \cdot \nabla \lambda \quad (\text{II.A.6})$$

$$\nabla \cdot \mathbf{D} \cdot \nabla \lambda = -\nabla \cdot \vec{u}^0, \quad (\text{II.A.7})$$

but the boundary conditions now contain contributions from the new surface integrals as well as from the integration by parts

$$\begin{aligned} 0 = & \int_S \lambda \vec{n} \cdot \delta \vec{u} dS - \int_{S_{\text{terrain}}} \lambda_{\text{terrain}} \vec{n} \cdot \delta \vec{u} dS + \int_{S_{\text{flow}}} f(S) (\vec{u} - \vec{u}^0) \cdot \delta \vec{u} dS \\ = & \int_{S_{\text{terrain}}} (\lambda_{\text{terrain}} - \lambda) \vec{n} \cdot \delta \vec{u} dS + \int_{S_{\text{flow}}} \left[ f(S) (\vec{u} - \vec{u}^0) + \lambda \vec{n} \right] \cdot \delta \vec{u} dS. \end{aligned} \quad (\text{II.A.8})$$

For the first integral to vanish,  $\lambda$  must be equal to  $\lambda_{\text{terrain}}$  on  $S_{\text{terrain}}$ . The flow-through boundary condition and terrain surface constraint can then be written as

---

<sup>1</sup> The original MATHEW code had OPEN boundaries corresponding to the ADJUST\_NORMAL condition and CLOSED boundaries equivalent to FIXED\_NORMAL on terrain surfaces with  $\vec{n} \cdot \vec{u}^0|_{S_{\text{terrain}}} = 0$ .

$$\begin{aligned}
\lambda + f(S)\bar{n} \cdot (\bar{u} - \bar{u}^0) &= \lambda + f(S)\bar{n} \cdot \mathbf{D} \cdot \nabla \lambda = 0 & S_{flow} \\
\bar{n} \cdot \bar{u} &= \bar{n} \cdot \bar{u}^0 + \bar{n} \cdot \mathbf{D} \cdot \nabla \lambda = 0 & S_{terrain}
\end{aligned}
\tag{II.A.9}$$

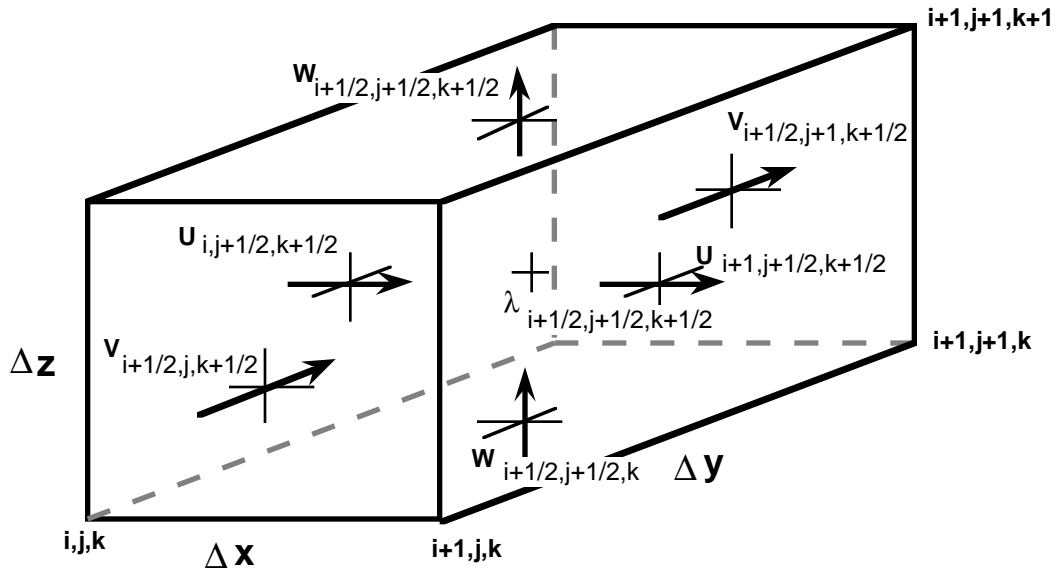
where the Euler-Lagrange equations have been used to substitute for  $\bar{u}$ . For a rectangular grid, these equations reduce to

$$\begin{aligned}
\lambda + \sigma_n^2 f(S) \frac{\partial \lambda}{\partial n} &= 0 & S_{flow} \\
\bar{n} \cdot \bar{u}^0 &= -\sigma_n^2 \frac{\partial \lambda}{\partial n} & S_{terrain}
\end{aligned}
\tag{II.A.10}$$

where  $\bar{n}$  is the *outward* normal to the face. Although in principle,  $f(S)$  could be any continuous function, in practice, it is taken to be a constant  $1/c_n$  governing the ratio of normal to tangential adjustment on each face, in order to limit the danger of overly constraining the problem. The ADJUST\_NORMAL and FIXED\_NORMAL boundary conditions are limiting cases of the MIXED boundary condition as  $c_n \rightarrow \infty$  or  $c_n \rightarrow 0$ , respectively.

### II.A.3. Derivation of the CG-MATHEW stencil

The CG-MATHEW grid is rectangular with regular spacings of  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$ , indexed by  $i$ ,  $j$ , and  $k$ , which run from 1 to  $i_{\max}$ ,  $j_{\max}$ , and  $k_{\max}$ , respectively. The wind field components are defined on staggered face centers, interpolated from input grid-point winds, and  $\lambda$  is defined at the cell center as shown in Figure II.A.1. Standard two- and three-point finite difference formulae then give the first derivatives of  $\lambda$  on the staggered face centers and both the second  $\lambda$  derivative and the wind component first derivatives at the cell centers. The topography is specified by cell-centered heights indexed by  $(i + \frac{1}{2}, j + \frac{1}{2})$ .



**Figure II.A.1:** Schematic representation of CG-MATHEW cell centered at point  $(i + 1/2, j + 1/2, k + 1/2)$  showing staggered face-centered wind and cell-centered  $\lambda$  components.

For computational purposes, the Poisson equation II.A.7 is divided through by  $\sigma_H^2$  and discretized, resulting in a set of linear equations in the components  $\lambda_{i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2}}$ , which can be written in matrix form as  $\mathbf{C}\vec{\lambda} = \vec{\mathbf{b}}$ .  $\vec{\lambda}$  and  $\vec{\mathbf{b}} = -(\nabla \cdot \vec{u}_0)/\sigma_H^2$  are vectors of length  $N = (i_{\max} - 1) \cdot (j_{\max} - 1) \cdot (k_{\max} - 1)$ , and  $\mathbf{C}$  is an  $N \times N$  matrix containing the finite-difference coefficients for each cell, which depend upon the boundary condition. Since the upper boundary of a given cell is the lower boundary of the next higher index cell on that axis,  $\mathbf{C}$  is symmetrical, and can be constructed to be positive definite ( $\vec{a} \cdot \mathbf{C} \cdot \vec{a} \geq 0$  for all vectors  $\vec{a}$ )<sup>2</sup>.

The stencil for the coefficient matrix can be derived for any one of the three axis directions, with symmetry used to establish the other components. In the  $x$  direction, the boundary conditions on the  $i$  and  $i + 1$  faces determine relationships between  $\lambda_{i-1/2}$ ,  $\lambda_{i+1/2}$ , and  $\lambda_{i+3/2}$ , which are used in the discretized Poisson and Euler-Lagrange equations

$$\frac{\lambda_{i+3/2} - 2\lambda_{i+1/2} + \lambda_{i-1/2}}{(\Delta x)^2} = -\frac{1}{\sigma_H^2} \frac{u_{i+1}^0 - u_i^0}{\Delta x}, \quad u_i = u_i^0 + \frac{\lambda_{i+1/2} - \lambda_{i-1/2}}{\Delta x}. \quad (\text{II.A.11})$$

Terrain cells are decoupled by setting the diagonal  $\lambda_{i+1/2}$  element to one and the off-diagonal elements of the matrix row to zero, so that the value of  $\lambda$  for a terrain cell is forced to be zero. Bordering interior cells impose no constraints on the  $\lambda$  values, while other boundary conditions restrict the allowed values as summarized in Table II.A.1. The opposite sign on the derivative for the  $i$  MIXED boundary condition reflects the reverse direction of the *outward* normal. It should be noted that for ADJUST\_NORMAL boundary conditions, the tangential component is algebraically constrained but not fixed, due to the staggered face-centered representation of the wind components. Combinations of the boundary conditions can be treated by a straightforward extension.

CELL $i + \frac{1}{2}$	BOUNDARY $i$	BOUNDARY $i + 1$	
terrain			$\lambda_{i+1/2} = 1$
interior	ADJUST_NORMAL	ADJUST_NORMAL	$\lambda_{i-1/2} = -\lambda_{i+1/2}$ $\lambda_{i+3/2} = -\lambda_{i+1/2}$
	FIXED_NORMAL	FIXED_NORMAL	$\lambda_{i-1/2} = \lambda_{i+1/2}$ $\lambda_{i+3/2} = \lambda_{i+1/2}$
	MIXED	MIXED	$\lambda_{i-1/2} = \left(\frac{1-c_x^+ \frac{\Delta x}{2}}{1+c_x^+ \frac{\Delta x}{2}}\right) \lambda_{i+1/2}$ $\lambda_{i-3/2} = \left(\frac{1-c_x^+ \frac{\Delta x}{2}}{1+c_x^+ \frac{\Delta x}{2}}\right) \lambda_{i+1/2}$

Table II.A.1: Boundary condition restrictions on  $\lambda$  values. The constant  $c_x^\pm$  controls the ratio of normal to horizontal adjustment for the MIXED condition.

---

<sup>2</sup> The original CG-MATHEW stencil was derived using quadratic differencing formulae leading to an asymmetric coefficient matrix.



Due to the symmetric sparse nature of the Poisson matrix, considerable memory savings are possible by storing only the diagonal and sub-diagonal elements. For the three-dimensional case, this can be generalized to storing the four possibly non-zero lower triangle components of  $\mathbf{C}$  in a compact  $4 \times N$  form written schematically as

$$\mathbf{A} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ C_{i,j,k-1} & C_{i,j,k} & C_{i-1,j,k} & C_{i,j,k} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad (\text{II.A.12})$$

where  $N$  is the number of cells. The  $n^{\text{th}}$  or  $(i, j, k)^{\text{th}}$  element of the matrix multiply  $\mathbf{C} \cdot \lambda$  then becomes

$$\begin{aligned} (\mathbf{C} \cdot \vec{\lambda})_n &= (\mathbf{C} \cdot \vec{\lambda})_{i,j,k} \\ &= A_{i,j,k,4} \lambda_{i,j,k} + A_{i,j,k,3} \lambda_{i-1,j,k} + A_{i,j,k,2} \lambda_{i,j-1,k} + A_{i,j,k,1} \lambda_{i,j,k-1} \\ &\quad + A_{i+1,j,k,3} \lambda_{i+1,j,k} + A_{i,j+1,k,2} \lambda_{i,j+1,k} + A_{i,j,k+1,1} \lambda_{i,j,k+1} \\ &= A_{n,4} \lambda_n + A_{n,3} \lambda_{n-1} + A_{n,2} \lambda_{n-i_{\max}} + A_{n,1} \lambda_{n-i_{\max} \cdot j_{\max}} \\ &\quad + A_{n+1,3} \lambda_{n+1} + A_{n+i_{\max},2} \lambda_{n+i_{\max}} + A_{n+i_{\max} \cdot j_{\max},1} \lambda_{n+i_{\max} \cdot j_{\max}} \end{aligned} \quad (\text{II.A.13})$$

since  $\pm i_{\max}$  moves  $\pm 1$  in the  $j$  coefficient and  $\pm(i_{\max} \cdot j_{\max})$  moves  $\pm 1$  in the  $k$  coefficient. We will also call this the matrix multiply  $\mathbf{A} \cdot \lambda$ .

The complete CG-MATHEW stencil in three-dimensions is given in terms of the components of  $\mathbf{A}$  in Table II.A.2. The matrix elements are found by summing together the contributions in each column which match the appropriate boundary conditions. For example, if the cell  $n = (i, j, k)$  is bordered by terrain cells at  $i - 1$  and  $j + 1$  and by interior cells on the remaining sides,  $A_{n,1} = \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$ ,  $A_{n,2} = \frac{1}{(\Delta y)^2}$ ,  $A_{n,3} = 0$ , and  $A_{n,4} = -\frac{1}{(\Delta x)^2} - \frac{1}{(\Delta y)^2} - \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{2}{(\Delta z)^2}$ .

#### II.A.4. Conjugate Gradient Algorithm

Conjugate gradient methods for solving an  $N \times N$  linear system  $\mathbf{A}\vec{x} = \vec{b}$  have several advantageous features. They require no parameter estimations and provide optimal error minimization. The algorithms have a finite termination after  $M \leq N$  steps, where  $M$  is the number of distinct eigenvalues of  $\mathbf{A}$ . Since they reference  $\mathbf{A}$  only through vector multiplication, the codes are also highly vectorizable and allow concise storage for sparse systems.

Conjugate gradient algorithms (Varga, 1962) are based on the idea of minimizing a function  $f(\vec{x}) = \frac{1}{2} \vec{x}^T \cdot \mathbf{A} \cdot \vec{x} - \vec{x}^T \cdot \vec{b}$  by finding the point where  $\nabla f = \mathbf{A} \cdot \vec{x} - \vec{b} = 0$ . The minimization is carried out by generating a succession of search directions  $\vec{p}_k$  and a corresponding set of improved solutions  $\vec{x}_{k+1}$  which minimize  $f$  in the space spanned by  $\{\vec{p}_1, \vec{p}_2, \dots, \vec{p}_k\}$ . Convergence is monotonic, the generalized error function  $(A\vec{x} - \vec{z})^T \cdot (\vec{x} - \vec{z})$  taking on its smallest value in the subspace spanned by the search when  $\vec{z} = \vec{x}_{k+1}$  and decreasing in magnitude with each iteration.

CELL	BOUNDARY		$A_{n,1}$	$A_{n,2}$	$A_{n,3}$	$A_{n,4}$
terrain	all boundaries					1
interior	all boundaries					$-\frac{2}{(\Delta x)^2} - \frac{2}{(\Delta y)^2} - \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{2}{(\Delta z)^2}$
	$x_{i-1}$	interior			$\frac{1}{(\Delta x)^2}$	
		$\lambda = 0$				$-\frac{1}{(\Delta x)^2}$
		$\frac{\partial \lambda}{\partial x} = 0$				$\frac{1}{(\Delta x)^2}$
		$c_x^- \lambda - \frac{\partial \lambda}{\partial x} = 0$				$\frac{1-c_x^- \Delta x / 2}{1+c_x^- \Delta x / 2} \cdot \frac{1}{(\Delta x)^2}$
	$x_{i+1}$	interior				
		$\lambda = 0$				$-\frac{1}{(\Delta x)^2}$
		$\frac{\partial \lambda}{\partial x} = 0$				$\frac{1}{(\Delta x)^2}$
		$c_x^+ \lambda + \frac{\partial \lambda}{\partial x} = 0$				$\frac{1-c_x^+ \Delta x / 2}{1+c_x^+ \Delta x / 2} \cdot \frac{1}{(\Delta x)^2}$
	$y_{j-1}$	interior		$\frac{1}{(\Delta y)^2}$		
		$\lambda = 0$				$-\frac{1}{(\Delta y)^2}$
		$\frac{\partial \lambda}{\partial y} = 0$				$\frac{1}{(\Delta y)^2}$
		$c_y^- \lambda - \frac{\partial \lambda}{\partial y} = 0$				$\frac{1-c_y^- \Delta y / 2}{1+c_y^- \Delta y / 2} \cdot \frac{1}{(\Delta y)^2}$
	$y_{j+1}$	interior				
		$\lambda = 0$				$-\frac{1}{(\Delta y)^2}$
		$\frac{\partial \lambda}{\partial y} = 0$				$\frac{1}{(\Delta y)^2}$
		$c_y^+ \lambda + \frac{\partial \lambda}{\partial y} = 0$				$\frac{1-c_y^+ \Delta y / 2}{1+c_y^+ \Delta y / 2} \cdot \frac{1}{(\Delta y)^2}$
	$z_{k-1}$	interior	$\left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$			
		$\lambda = 0$				$-\left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$
		$\frac{\partial \lambda}{\partial z} = 0$				$\left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$
		$c_z^- \lambda - \frac{\partial \lambda}{\partial z} = 0$				$\frac{1-c_z^- \Delta z / 2}{1+c_z^- \Delta z / 2} \cdot \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$
	$z_{k+1}$	interior				
		$\lambda = 0$				$-\left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$
		$\frac{\partial \lambda}{\partial z} = 0$				$\left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$
		$c_z^+ \lambda + \frac{\partial \lambda}{\partial z} = 0$				$\frac{1-c_z^+ \Delta z / 2}{1+c_z^+ \Delta z / 2} \cdot \left(\frac{\sigma_V}{\sigma_H}\right)^2 \frac{1}{(\Delta z)^2}$

Table II.A.2: CG-MATHEW stencil. The values of the stencil for any cell  $n$  are found by summing contributions from all applicable rows.

For improved convergence, preconditioning is used to modify the matrix  $\mathbf{A}$  to one that is very close to the identity matrix  $\mathbf{I}$  except for a few extreme eigenvalues. This is done by multiplying  $\mathbf{A}$  by a easily calculable sparse symmetric matrix  $\tilde{\mathbf{A}}^{-1}$

$$\begin{aligned}\tilde{\mathbf{A}}^{-1} \mathbf{A} &\approx \mathbf{I} \\ \tilde{\mathbf{A}}^{-1} \mathbf{A} \vec{x} &= \tilde{\mathbf{A}}^{-1} \vec{b} .\end{aligned}\tag{II.A.14}$$

As a first approximation, diagonal scaling

$$\tilde{\mathbf{A}}_{i,j}^{-1} = \frac{1}{\mathbf{A}_{i,j}}\tag{II.A.15}$$

is used for preconditioning.

The numerical algorithm (Press et al., 1993) uses an initial guess for  $\vec{x}_1$ , which is an arbitrary approximation to the solution vector and may be taken to be the zero vector. The first residual  $\vec{r}_1$ , the preconditioned residual  $\vec{z}_1$ , and an initial direction vector  $\vec{p}_1$  are generated according to

$$\begin{aligned}\vec{r}_1 &= (\vec{b} - \mathbf{A} \vec{x}_1) \\ \vec{z}_1 &= \tilde{\mathbf{A}}^{-1} \vec{r}_1 \\ \vec{p}_1 &= \vec{z}_1 .\end{aligned}\tag{II.A.16}$$

The minimization then proceeds for successive  $k$  by finding the  $\alpha_k$  that minimize  $f(\vec{x}_k + \alpha_k \vec{p}_k)$  and generating the next search direction vector by  $\mathbf{A}$ -orthogonalization of the residuals according to the prescription

$$\begin{aligned}\alpha_k &= \frac{\vec{r}_k \cdot \vec{z}_k}{\vec{p}_k \cdot \mathbf{A} \cdot \vec{p}_k} \\ \vec{r}_{k+1} &= \vec{r}_k - \alpha_k \cdot \mathbf{A} \cdot \vec{p}_k \\ \vec{z}_{k+1} &= \tilde{\mathbf{A}}^{-1} \cdot \vec{r}_{k+1} \\ \vec{x}_{k+1} &= \vec{x}_k + \alpha_k \vec{p}_k \\ \beta_k &= \frac{\vec{r}_k \cdot \vec{z}_{k+1}}{\vec{r}_k \cdot \vec{z}_k} \\ \vec{p}_{k+1} &= \vec{z}_k - \beta_k \vec{p}_k .\end{aligned}\tag{II.A.17}$$

The residuals and direction vectors are mutually orthogonal and conjugate

$$\begin{aligned}\vec{r}_i \cdot \vec{r}_j &= 0 \quad j < i \\ \vec{r}_i \cdot \vec{p}_j &= 0 \quad j \leq i \\ \vec{p}_i \cdot \mathbf{A} \cdot \vec{p}_j &= 0 \quad j < i\end{aligned}\tag{II.A.18}$$

so that each iteration proceeds by making the residual orthogonal to one more independent conjugate direction vector. The residuals  $\vec{r}_k$  are also linearly independent and give the direction of steepest descent  $f(\vec{x})$  at each step. It should be noted that the conjugate direction vectors  $\vec{p}_k$  are *not* determined beforehand. Instead, they are constructed as linear combinations of the  $\vec{r}_k$  by calculation of the appropriate  $\beta_k$  at each iteration.

Any of several convergence criterion can be adopted depending on the application. We have chosen a combination of

$$\begin{aligned} \left| \mathbf{A} \vec{x}_k - \vec{b} \right| &< \Delta_{tolerance} \\ \frac{|\vec{x}_{k+1} - \vec{x}_k|}{|\vec{x}_{k+1}|} &< \Delta_{tolerance} \end{aligned} \tag{II.A.19}$$

The first tolerance check is used to stop the iterations, whenever the second condition is satisfied. These criteria are not completely robust; the final field should only be trusted when the solution converges monotonically.

### II.A.5. Output

The face-centered adjusted velocity components ( $u$ ,  $v$ ,  $w$ ) are interpolated back to form grid-point winds using an averaging procedure which maintains mass-consistency. The grid-point velocities are then written to MATVEL files (see Section III.C.8) along with traceback black information which uniquely identifies the current problem. These files provide input for the pollutant transport and diffusion code ADPIC.

## II.B. ADPIC

### II.B.1. Formulation—Pseudo-Velocity and particle-in-cell method

The pseudo-velocity method consists of the following: consider the non-linear transport-diffusion equation

$$\frac{\partial \chi}{\partial t} + \underline{U} \cdot \nabla \chi = \nabla \cdot (K \nabla \chi) , \quad (\text{II.B.1})$$

where  $\chi$  is a scalar concentration,  $K$  the diffusion coefficient and  $\underline{U}$  the (given) mass-consistent wind advection field (Sherman, 1978b). Under the assumption of incompressibility we can replace the  $\underline{U} \cdot \nabla \chi$  term by  $\nabla \cdot (\chi \underline{U})$ . Upon combining the two divergence terms, we can rewrite Eq. (II.B.1) in its flux conservative (pseudo velocity) form

$$\frac{\partial \chi}{\partial t} + \nabla \cdot \chi \left( \underline{U} - \frac{K}{\chi} \nabla \chi \right) = \frac{\partial \chi}{\partial t} + \nabla \cdot (\chi \underline{U}_p) = 0 \quad (\text{II.B.2})$$

where  $\underline{U}_p = \underline{U} - \frac{K}{\chi} \nabla \chi$  is the pseudo transport velocity.

The particle-in-cell method divides the pollutant into discrete mass particles representing the source in terms of a large number of “numerical” pollutant particles, which are transported and dispersed in the Lagrangian sense within a fixed Eulerian domain (Lange, 1973, 1978; Welch *et al.*, 1965; Amsden, 1966; Sklarew, 1971).

The grid mesh of the code is represented by an Eulerian grid consisting of three-dimensional rectangular cells of uniform size. The concentrations  $\chi$  are defined at the centers of the cells and the velocities  $\underline{U}$ ,  $\underline{U}_p$  and  $-\frac{K}{\chi} \nabla \chi$  are defined at the cell corners. The locations of the particles, which represent the pollutant cloud, are defined by their individual Lagrangian coordinates within the Eulerian fixed grid.

A time cycle of the code is divided into an Eulerian step and a Lagrangian step and proceeds as follows:

- *Eulerian Step*: The concentrations  $\chi$ , given for each cell at the beginning of the cycle, are used to calculate the diffusivity velocities  $\underline{U}_D = -\frac{K}{\chi} \nabla \chi$  which are then added to the given wind advection velocities  $\underline{U}$  to yield a pseudo-velocity  $\underline{U}_p$  for each cell corner.
- *Lagrangian Step*: Each marker particle contained in a given cell is transported for one time step  $\Delta t$  with a velocity  $\underline{U}_p$ , which is computed from the pseudo-velocities  $\underline{U}_p$  at the corners of the cell, and the particle coordinates  $\underline{x}$  are determined by a volume weighting scheme

$$\underline{x}(\text{new}) = \underline{x}(\text{old}) + \underline{U}_p \Delta t . \quad (\text{II.B.3})$$

Finally, a new concentration distribution  $\chi$  is calculated from the new particle positions, thus ending the cycle.

The chief advantage of using such a hybrid Eulerian-Lagrangian scheme is that the fictitious diffusion inherent in a purely Eulerian scheme is eliminated. The truncation errors inherent in the finite difference algorithms remain, of course, and must be minimized by the choice of the time step.

## **II.B.2. Boundary Conditions**

The boundary conditions are applied to a layer of boundary cells that surround or protrude into the grid in the form of topography.

### **II.B.2.a. Topography**

Space-variable topography protrudes into the basic grid block of the code. The code recognizes individual flagged grid cells as topography and uses the applicable boundary conditions. The proper modeling of particle deposition on the ground is greatly dependent on the way topography is treated. Dry deposition of small particles and deposition of larger particles by gravity are treated by specifying a deposition velocity at the topographic surface, and a Stoke/McDonald-type fall velocity, respectively. This surface boundary condition is the same for CG-MATHEW and is supplied by the TOPOG routine.

### **II.B.2.b. Flux Boundary Conditions**

Flux boundary conditions representing inflow or outflow of pollutant particles from the grid are cast in the form of conservation of total flux, i.e.,  $(\underline{U}_p \chi) = \text{constant}$  and  $(\underline{U}_p \chi) = 0$  at the surface [see Eq. (II.B.2)]. An inversion layer capping the region of interest can be specified by a reflecting boundary condition, making the total flux  $(\underline{U}_p \chi) = 0$  at the inversion layer.

Due to the part-Eulerian, part-Lagrangian makeup of ADPIC, one set of boundary conditions is imposed on the Eulerian velocity field and another set is imposed on the Lagrangian particles. Both sets must be consistent with each other.

The two basic boundary conditions imposed on the pseudo-velocity field in ADPIC  $(\underline{U}_p = \underline{U} + \underline{U}_D)$  are constant mass flow,  $(\chi \underline{U}) = \text{constant}$ , corresponding to inflow and outflow of particles, and zero mass flow  $(\chi \underline{U}) = 0$ , corresponding to reflection of particles from the boundary. There are intermediate cases as, for example, deposition of particles at the topography, in which case a deposition velocity is specified. The chief cause of error incurred by velocity boundary conditions is the inability to specify the variability of the outflow boundary conditions. The concentration field is generally smooth enough by the time it reaches the outflow boundary that the boundary velocities can be specified by postulating a constant flux of particles through the boundary grid cell layer.

The particle boundary conditions are very simple. If a particle leaves the grid during a cycle, it is either destroyed, counted as deposited, or reflected, according to the type of boundary specified.

### II.B.3. Diffusion Parameters

Because ADPIC is a three-dimensional Cartesian code and uses the turbulent flux diffusivity pseudo-velocity

$$\underline{U}_D = -\frac{K}{\chi} \nabla \chi, \quad (\text{II.B.7})$$

it can, in principle, accommodate the full Cartesian eddy diffusivity tensor  $\mathbf{K} = k_{ij}$ . In practice, isotropy of turbulence in the atmosphere is assumed, and only the diagonal terms  $K_{xx}$ ,  $K_{yy}$ , and  $K_{zz}$  survive. In addition, it is generally assumed that the horizontal components of the diffusivity coefficients are the same, so that  $K_{xx} = K_{yy}$ . Even with these simplifications, the capability of ADPIC to model diffusion in three-dimensions with topography and inversions, under conditions of wind shear and with space-and time-variable diffusivity coefficients, cannot presently be fully exploited largely because of the lack of a unified theory of diffusion in the atmosphere from which such diffusion coefficients can be obtained. Consequently, the partially empirical theories of diffusion in the atmosphere which have been tested must be combined in a consistent way to provide diffusion coefficients for ADPIC.

For present purposes, diffusivity coefficients must be found that are valid for single puff or continuous plume releases which range initially from meters to kilometers. With time periods of about 12h and geographical radii of about 100 km, the turbulent diffusivities are scale-dependent. Additionally, since the pollutant cloud may extend vertically through the entire turbulent mixing layer, the atmospheric thermal stability must be taken into account. Constant K-theory is not adequate for such requirements, and similarity theories have been chosen as the primary available framework upon which to base horizontal and vertical diffusivity coefficients (Batchelor, 1947 and 1950). ADPIC computes both horizontal and vertical diffusivities,  $K_h$  and  $K_z$ , in a variety of forms based on accepted diffusion theory.

#### II.B.3.a. Horizontal Diffusion Coefficients

##### II.B.3.a.b. Continuous Sources

For continuous sources, the horizontal diffusivity  $K_h$  is based on the semi-empirical expression (Draxler, 1976)

$$\sigma_y = \sigma_\theta U t f\left(\frac{t}{\tau}\right); f\left(\frac{t}{\tau}\right) = \left(1 + \frac{t}{\tau}\right)^{-1/2} \quad (\text{II.B.8a})$$

combined with the analytical relationship

$$K_h = \sigma_y \frac{d\sigma_y}{dt}, \quad (\text{II.B.8b})$$

where  $\sigma_y$  is the horizontal standard deviation of the pollutant plume width,  $\sigma_\theta$  is the standard deviation of the fluctuation of the wind direction,  $U$  is the mean wind speed,  $t$  is time, and  $f(t, \tau)$  is

a function of time and an empirical time constant  $\tau$  which will assure that Eq. (II.B.8a) will go to the Taylor limit ( $\sigma_y \propto t^{1/2}$ ) for times  $t \gg \tau$ . Note that  $K_h$  is a function of height also, through the height dependence of  $\sigma_\theta$ .

### II.B.3.a.c. Instantaneous Clouds

A large number of experimental studies have been summarized (Crawford, 1966) to show the validity of similarity theory in the inertial subrange of the atmosphere for the horizontal spreading of clouds in the atmosphere from meter to kilometer sizes over times from minutes to several days. The conclusion drawn is that for such scales the horizontal diffusivity coefficients take the form

$$K_{xx} = K_{yy} \propto \varepsilon^{1/3} (\sigma_y^2)^{2/3}, \quad (\text{II.B.9})$$

where  $\varepsilon$  is the rate of eddy-energy transfer in the inertial range and is a scaled function of height  $Z$  and average horizontal windspeed  $U(Z)$  above the surface,

$$\varepsilon \propto \frac{[U(Z)]^3}{Z}, \quad (\text{II.B.10})$$

and  $\sigma_y^2$  is the average horizontal dispersion of the cloud. The expression for scale dependent diffusion (Walton, 1973)

$$\sigma_y^2 = \left( \sigma_0^{2/3} + \frac{2}{3} C \varepsilon^{1/3} t \right)^3 \quad (\text{II.B.11})$$

is substituted into Eq. (II.B.9) and provides  $K_{xx}$  and  $K_{yy}$  as a function of  $t$  and  $\varepsilon$ . For early times, when the initial cloud size in terms of its standard deviation  $\sigma_0$  is important, the time rate of change of the dispersion is proportional to time  $t$ ,

$$\frac{d(\sigma_y^2)}{dt} \propto (\varepsilon \sigma_0)^{2/3} t. \quad (\text{II.B.12})$$

At late times, when  $\sigma_0^2 \ll \sigma_y^2$ , the time rate of change of the dispersion becomes

$$\frac{d(\sigma_y^2)}{dt} \propto \varepsilon t^2. \quad (\text{II.B.13})$$

### II.B.3.d. Vertical Diffusivity Coefficient

For the vertical diffusivity coefficient  $K_{zz}$ , the turbulent diffusion due to both the mechanical shear induced by the drag of the earth's surface on the atmosphere and the vertical thermal convection within the atmosphere must be taken into account. Within the surface layer of the atmosphere, the original similarity concepts of Monin and Obukhov (1954), based on constant momentum and heat flux, are used to calculate diffusivities. The Monin-Obukhov similarity parameter  $\xi = Z/L$ , where  $L$  is the Monin-Obukhov length and  $Z$  is the vertical height, is the



primary independent variable representing the state of stability of the atmosphere in the surface layer. It is used to obtain the vertical diffusivity coefficient  $K_{zz}(\xi)$ .

$$K_{zz} = U_* k Z \frac{1}{\phi_m} , \quad (\text{II.B.14})$$

where  $U_*$  is the friction velocity,  $k$  is the von Karman constant,  $\phi_m = (k Z/U_*)(\partial U/\partial Z)$  is a dimensionless wind shear, and  $U$  is the average horizontal wind speed. The function  $\phi_m$  depends in the following way on the stability of the atmosphere:

$$\begin{aligned} \text{Stable:} \quad \phi_m &= 1 + \gamma_1 \xi , \\ \xi &> 0, L > 0 \end{aligned} \quad (\text{II.B.15})$$

$$\begin{aligned} \text{Neutral:} \quad \phi_m &= 1 , \\ \xi &= 0, L \rightarrow \infty \end{aligned} \quad (\text{II.B.16})$$

$$\begin{aligned} \text{Unstable:} \quad \phi_m &= (1 - \gamma_2 \xi)^{-1/4} , \\ \xi &< 0, L < 0 . \end{aligned} \quad (\text{II.B.17})$$

Here  $\gamma_1$  and  $\gamma_2$  are empirical constants ranging from 4 to 5 and 14 to 16, respectively (Businger *et al.*, 1971). The Obukhov length  $L$  has the form

$$L = - \frac{U_*^3 C_p \rho T}{kgH} , \quad (\text{II.B.18})$$

where  $C_p$  is the specific heat of air at constant pressure,  $\rho$  is the density of air,  $T$  is the temperature,  $H$  is the heat flux, and  $g$  is the gravitational acceleration. When  $L$  is not known, it can be found from its functional relationship to the Richardson number.

Above the surface layer and up to the turbulent mixing height, turbulence due to the surface stress eddies, thermal convection, and wind shear decreases with height until it essentially disappears at the top of the mixing layer (Holzworth, 1964). The similarity assumptions of constant momentum and heat flux are no longer valid in this region, and very little consensus exists among investigators at the present time as to the values of vertical diffusion parameters. This is largely due to the complex interactions of wind shear, lapse rate, Coriolis force, pressure gradients, eddy diffusivity, and buoyancy in this region. It is generally assumed that the vertical diffusivity coefficient  $K_{zz}$  either remains constant up to the top of the mixing or inversion height, where it sharply drops to a minimum value, or approaches a minimum based on a smooth functional dependence on height above the surface layer.

The vertical diffusivity  $K_z$  is computed in ADPIC separately for the atmospheric surface layer and the outer boundary layer. For the surface layer,  $K_z$  is based on similarity theory (Businger *et al.*, 1971)

$$K_z = \frac{k U_* z}{\phi(z/L)} e^{-\left(\frac{V_g z}{U_* h}\right)} \quad (\text{II.B.19})$$

where  $k$  is the von Karman constant,  $U_*$  is the friction velocity computed by the code,  $z$  is the height above terrain, and  $\phi(z/L)$  is the atmospheric stability function based on  $z$  and the Monin-Obukhov scale length  $L$ . In the outer atmosphere boundary layer,  $K_z$  is computed by the empirical form

$$K_z = CU_* z e^{-(z/h)^4} , \quad (\text{II.B.20})$$

where  $C$  is an empirically-derived proportionality coefficient (Start *et al.*, 1975; Crawford, 1974), and  $h$  is the mixing height of the atmosphere boundary layer.

Another reasonable value for  $K_z$  above the surface layer is a statistically obtained power law profile for the vertical standard deviation  $\sigma_z$  of the form

$$\sigma_z = a\sigma_\theta x^b , \quad (\text{II.B.21})$$

where  $a$  and  $b$  are observationally obtained stability dependent constants,  $\sigma_\theta$  is the horizontal wind fluctuation over a time interval, and  $x$  is the downwind distance from the source. From the relation between  $K$ -theory and statistical diffusion theory, the vertical diffusivity coefficient  $K_z$  then becomes

$$\frac{d[\sigma(t)]^2}{dt} = 2K_z , \quad (\text{II.B.22})$$

where  $t$  is the time of travel from the release point (Slade, 1968; Pasquill, 1971).

#### **II.B.4. Options in ADPIC**

##### **II.B.4.a. Source Parameters**

Up to MAXSOURC separate pollutant source configurations (combinations of source type, rate, geometry, etc.) are allowed. The release rate and source center height of each pollutant source can vary with time. The effective times of the time-varying source parameters (SOURCE\_RATE and CENTER\_HGT) are controlled by the timing parameters SOURCE\_START\_DATE and SOURCE\_START\_TIME.

The way in which ADPIC describes pollutants in terms of discrete mass (activity) particles makes the implementation of decay rate calculations straightforward. Such processes as radioactive decay of pollutants in conjunction with atmospheric dispersion under complex terrain conditions can be modeled.

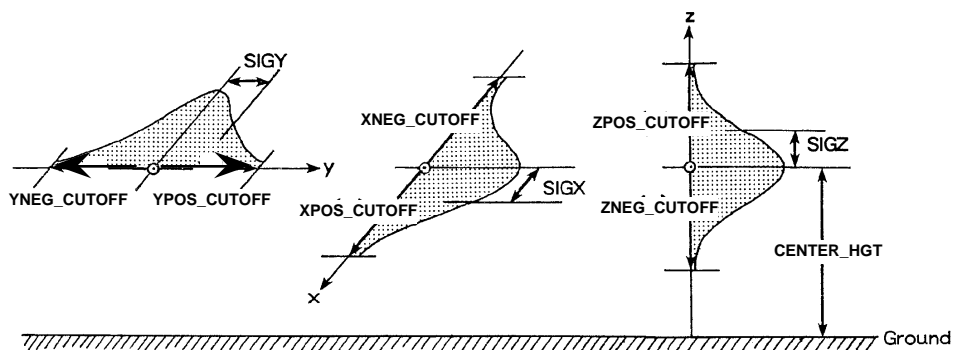
##### **II.B.4.b. Source Geometry**

The initial position ( $x$ ,  $y$ , and  $z$  coordinates) of each marker particle representing the pollutant material is computed by using a Gaussian or normal probability distribution function. The probability of the distance that each particle will be from each of the three orthogonal axes through

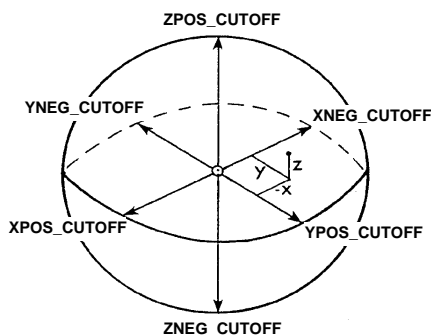
the source center is determined by the shape of a Gaussian distribution. The number of particles to be generated for each source is specified by the user by the use of the TOTPART parameter. Several thousand particles per ADPIC source provide a fairly good representation of a sample population.

In all, ten parameters are used to control the representation of the initial stabilized source geometry. The first three are standard deviations of the Gaussian distribution, SIGX, SIGY and SIGZ, in the  $x$ ,  $y$ , and  $z$  directions, respectively. These control how peaked or flat the source distribution will be along each axis, as shown in Fig. II.B.1. The Gaussian distributions are then truncated as shown in Fig. II.B.2 using six source cutoff parameters, XPOS\_CUTOFF, XNEG\_CUTOFF, YPOS\_CUTOFF, YNEG\_CUTOFF, ZPOS\_CUTOFF, and ZNEG\_CUTOFF, where POS is for right or top, NEG for left or bottom. The last source geometry parameter is the height above ground of the source center, CENTER\_HGT.

If very little truncation of the Gaussian curves is desired, the radius of the entire source is typically used for the cutoff values and the radius divided by 2.5 to compute the standard deviations. This is based on the fact that about 99% of the area in the Gaussian distribution lies within 2.5 standard deviations of its center.



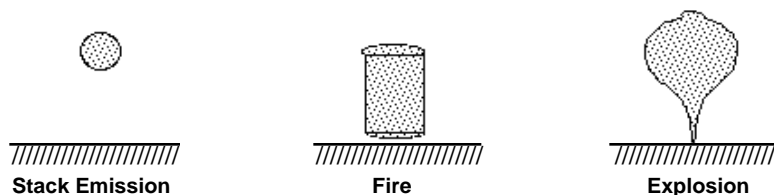
**Fig. II.B.1. Gaussian source geometry parameters**



**Fig. II.B.2. Cutoff parameters for a spherical-shaped source**

When the pollutant distribution covers fewer than two grid cell lengths, the concentration gradients are ill-defined and the code diffuses the particles too slowly, in which case a special sub-grid treatment of the source particles is employed for the diffusion only. The initial source parameters maintain a Gaussian shaped plume independently in the horizontal and vertical until those respective standard deviations become as large as the grid cell size. At that time, the diffusion scheme specified by the user (Gradient or Monte Carlo, controlled by the parameter DIFF\_METHOD ) takes over. If it is desired to extend Gaussian plume shape past this point, the ADPIC.NML parameters GAUSS\_DIFF\_MULT\_X and GAUSS\_DIFF\_MULT\_Z can be used to override the default time at which the diffusion scheme specified by IDIF replaces the extension of the Gaussian source geometry.

The ten source parameters may be used to describe a wide variety of sources, with the three most common shown in Fig. II.B.3. Releases from stacks are usually assumed to be spherical (equal sigma and cutoff values in each direction). Puffs or clouds from fires or explosions are usually elongated in the vertical. If necessary an explosion may be described by two source terms, one for the lower stem and one for the upper cloud.



**Fig. II.B.3. Example source geometries**

#### **II.B.4.c. Particle Size Distribution**

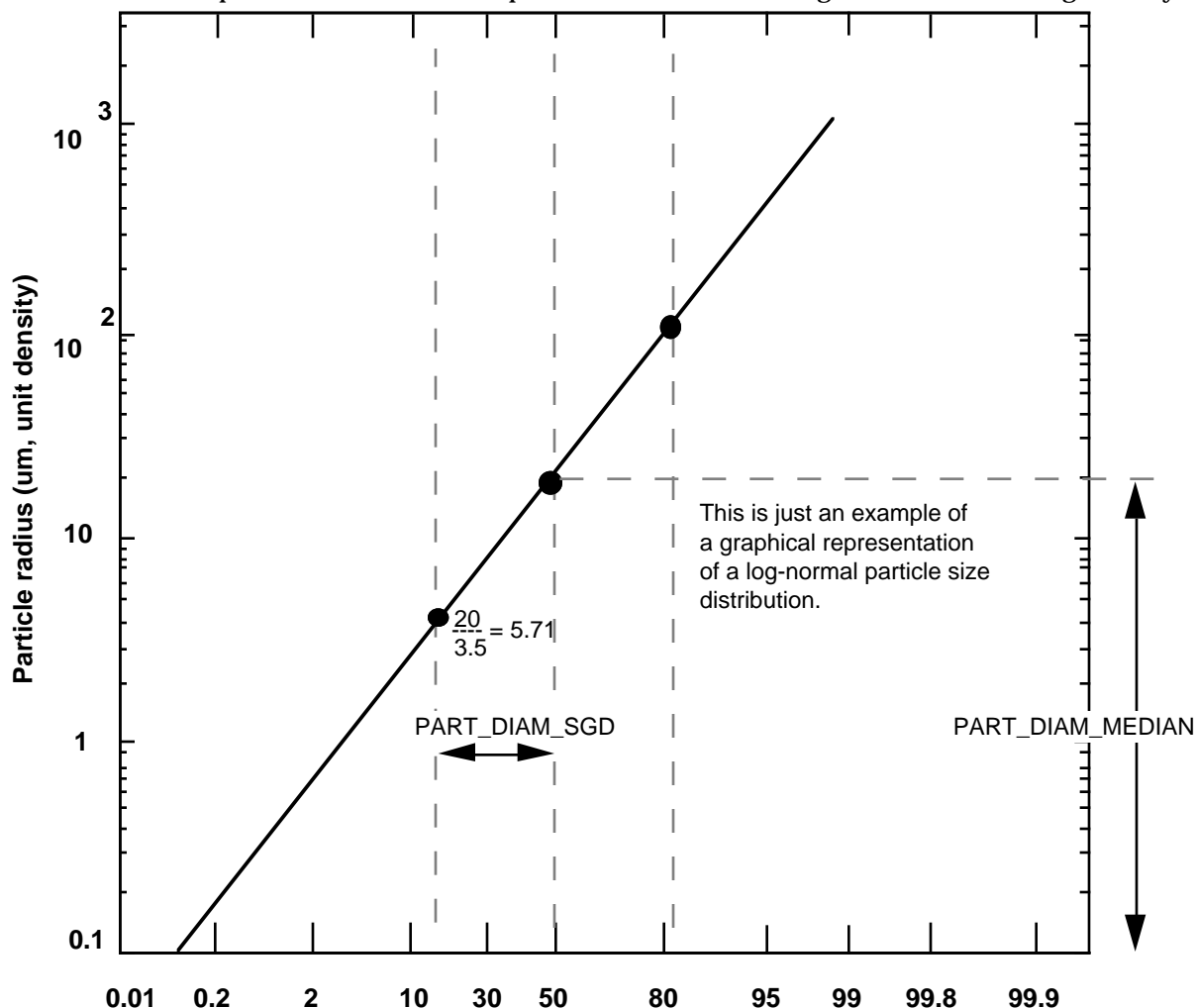
The distribution of the particle size range as a log-normal distribution can be specified in ADPIC by using the parameters PART\_DIAM\_MIN, PART\_DIAM\_MAX, PART\_DIAM\_SGD, and PART\_DIAM\_MEDIAN. An example showing the graphical derivation of these values is shown in Fig. II.B.4. Usually particle sizes are normalized to a particle density of 1 g/cm<sup>3</sup> (1000 kg/m<sup>3</sup>). If actual particle sizes are used, then the actual particle density (PART\_DENSITY) must be specified.

#### **II.B.4.d. Dry Deposition**

The removal of material from the atmosphere through dry deposition is modeled separately by ADPIC via two processes: gravitational settling of material, and flux of material from the surface layer to the ground surface itself (represented by the deposition velocity).

The settling velocity is modeled by ADPIC with either the Stokes law or the McDonald method (McDonald, 1960) based on particle size (PART\_DIAM\_MEDIAN), particle density (PART\_DENSITY), and altitude. The McDonald method is used for cases when the Reynold's number is greater than 1 and the Stokes Law is invalid (e.g. particle diameters greater than approximately

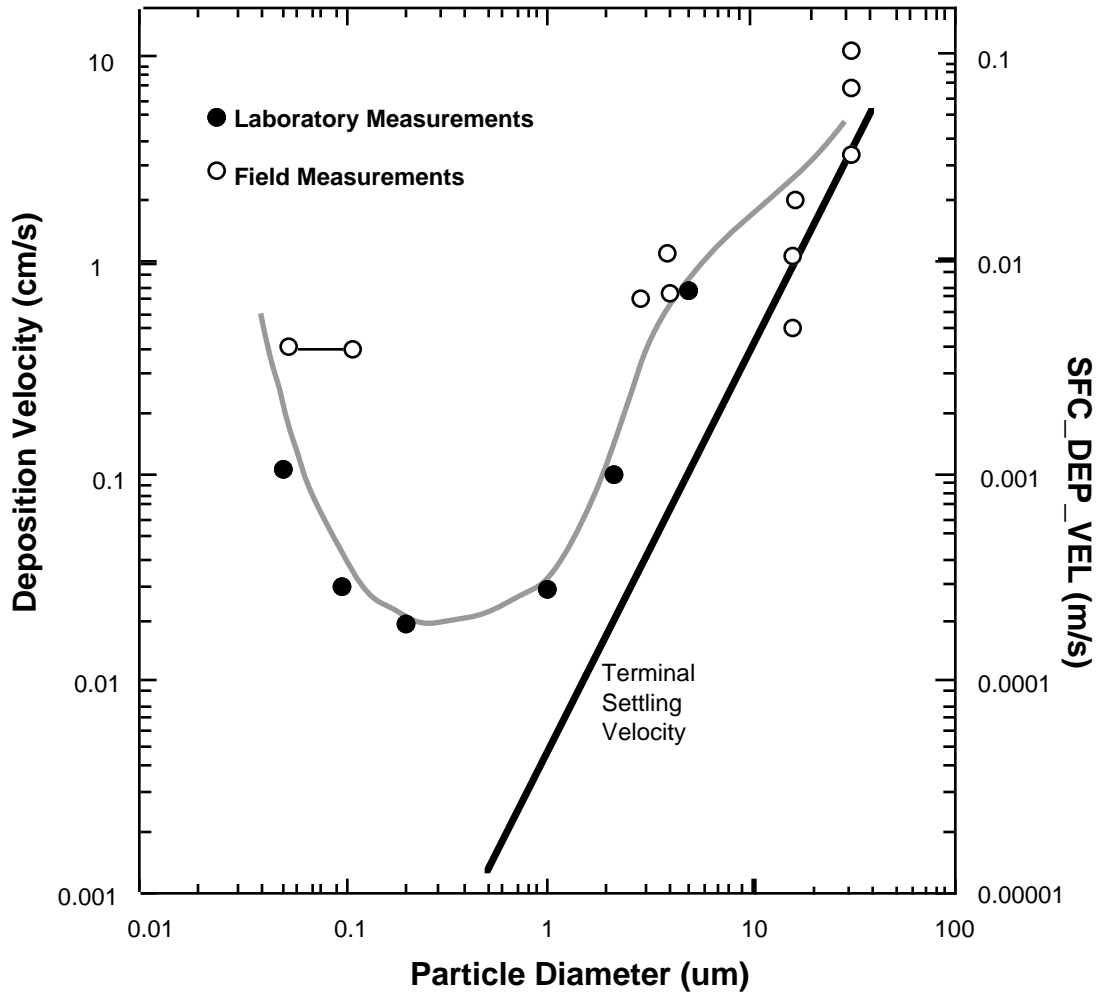
3.5  $\mu\text{m}$  at standard ground-level conditions). The parameters PART\_DIAM\_MEDIAN and PART\_DENSITY are used not only to specify the median particle size diameter and the particle density, respectively, but are also used as flags: if either PART\_DIAM\_MEDIAN or PART\_DENSITY is zero, the ADPIC particles are treated as passive material with no gravitational settling velocity.



Log-normal particle size parameters for ADPIC\_SOURCE namelist

Input parameters	Percent less than stated size	Description
PART_DIAM_SGD		Standard geometric deviation: the ratio of radius 50% or radius 84% radius 16% or radius 50%
PART_DIAM_MEDIAN ( $\mu\text{m}$ )		Median diameter: diameter of 50% point
PART_DIAM_MAX ( $\mu\text{m}$ )		Maximum diameter of particles
PART_DIAM_MIN ( $\mu\text{m}$ )		Minimum diameter of particles

Fig. II.B.4. Log normal particle size distribution



**Fig. II.B.5. Laboratory and Field Measurements of Deposition Velocity to Grass vs. Particle Size (McMahon and Dennison, 1979)**

The surface deposition velocity, which is modeled separately from the gravitational settling velocity, represents the affinity of the material for the ground. The deposition velocity is typically a function of effluent chemistry, particle size, atmospheric stability, and surface type (e.g. soil, vegetation, etc.). Fig. II.B.5 shows how deposition velocity varies with particle diameter in laboratory and field measurements made by McMahon and Dennison (1979).

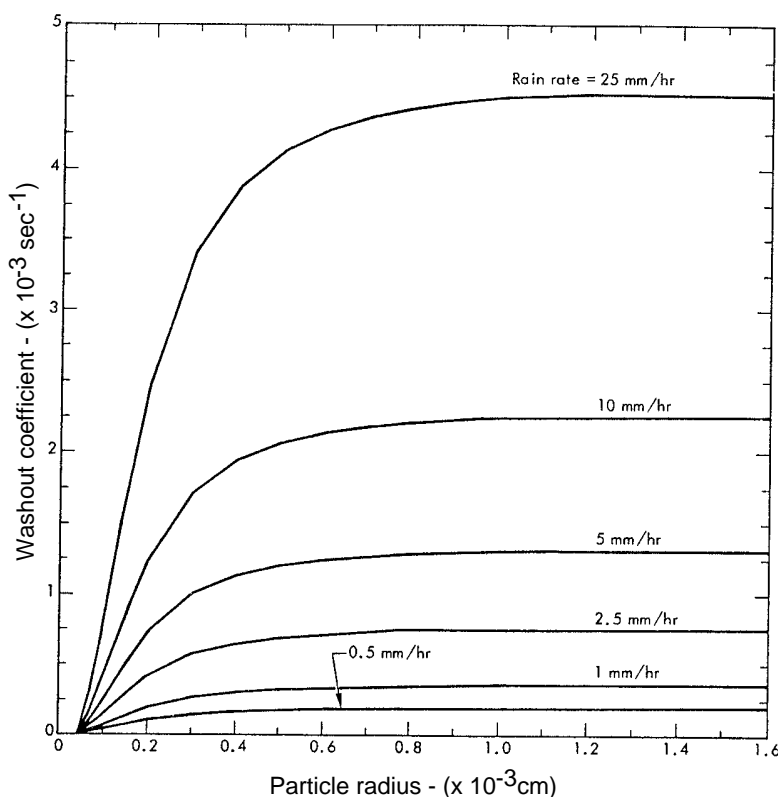
The deposition velocity used by ADPIC is linearly weighted according to the distance from ground level, from zero at the height specified by SFC\_DEP\_HGT to the full value of SFC\_DEP\_VEL at ground level. A non-zero deposition velocity will cause an ADPIC particle to be deposited if it is at a height which is less than or equal to the weighted SFC\_DEP\_VEL (in m/sec) times the time step (in sec). A SFC\_DEP\_VEL value of zero will result in no removal of ADPIC particles via the deposition velocity mechanism.

If either PART\_DIAM\_MEDIAN or PART\_DENSITY is zero, and if SFC\_DEP\_VEL are zero, the ADPIC particles are treated as passive material and no dry deposition will result. Although material can be moved into the surface layer from above via advection by the mean wind and by turbulent diffusion, material cannot be moved from the surface layer to the ground through either of these two methods as modeled by ADPIC. Any material that impacts the ground through either advection or diffusion will be reflected from the ground.

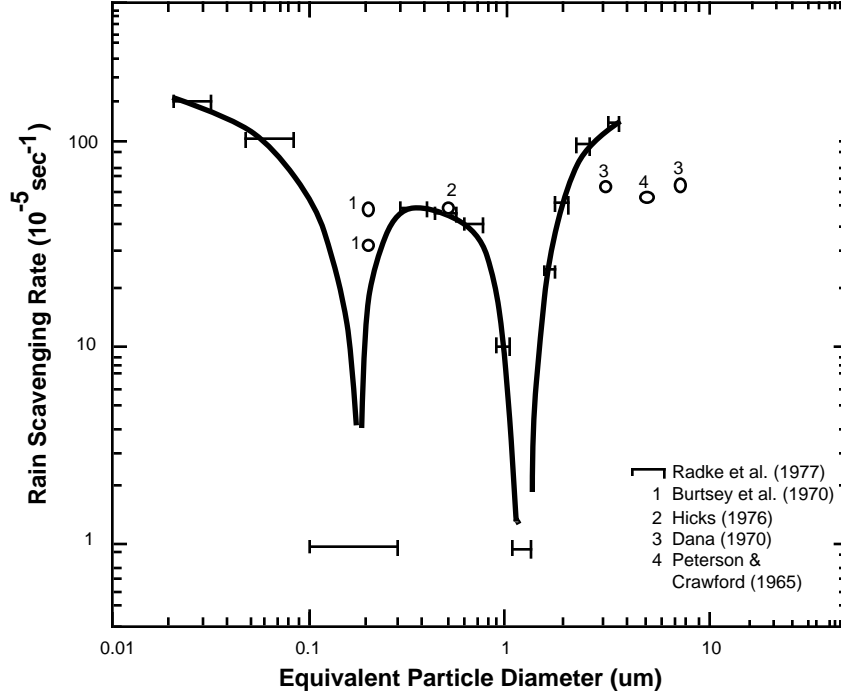
#### II.B.4.e. Washout

At present, the basic assumption for model removal of debris particles in ADPIC is that the base of the rain cloud is above the debris cloud. For surface or near-surface pollutant releases, this seems to be a reasonable assumption to about 100 km downwind. Thus, ADPIC in its present form does not treat in-cloud rainout, but has provisions for below-cloud washout (Crandall et al, 1973).

Washout is treated in the following way in ADPIC. For a specified rain rate over a given surface area, a washout coefficient  $A(r)$  is assumed as a function of particles of radius  $r$  (see Figs. II.B.6 and II.B.7).



**Fig. II.B.6. Washout coefficient versus particle radius for various rain rates. (Crandall *et al.*, 1973)**



**Fig. II.B.7. Measured Values of Scavenging Coefficient vs. Particle Size (McMahon and Dennison, 1979)}**

Because of their limited number, each ADPIC particle is only a representation of a certain amount of an actual pollutant. Therefore, an ADPIC mass particle is initially considered to represent an integrated pollutant particle size distribution over sizes such that

$$M(0) = \int_{r=0}^{\infty} m(r, 0) dr \quad , \quad (\text{II.B.4})$$

where  $M(0)$  is the initial mass of an ADPIC particle and  $m(r, 0)$  is the mass per unit radius of a pollutant particle of radius  $r$  at time 0. Constant density is assumed for the pollutant particles.  $M(t)$ , the mass of an ADPIC particle remaining after a time  $t$  in the rain for a given constant rain rate and raindrop size distribution, can be written as

$$\begin{aligned} M(t) &= \int_{r=0}^{\infty} m(r, t) dr \\ &= \int_{r=0}^{\infty} m(r, 0) e^{-\Lambda(r)t} dr \quad . \end{aligned} \quad (\text{II.B.5})$$

To account for the diffusion and transport of pollutants as calculated by ADPIC, this function  $M(t)$  is obtained from a table at each ADPIC time step  $\Delta t$ . The fraction of mass  $F(\Delta t)$  of an ADPIC particle washed out and deposited on the surface during one ADPIC time cycle then becomes

$$F(\Delta t) = \frac{M(t) - M(t + \Delta t)}{M(t)} = 1 - \frac{\int_{r=0}^{\infty} m(r, 0) e^{-\Lambda(r)(t-\Delta t)} dr}{\int_{r=0}^{\infty} m(r, 0) e^{-\Lambda(r)t} dr} \quad . \quad (\text{II.B.6})$$

This fraction of mass loss per cycle for each ADPIC particle is stored cumulatively each cycle in a two-dimensional, surface deposition array.



#### **II.B.4.f. Plume Rise**

Plume rise can significantly affect ground level concentrations from stack releases into the atmosphere. It becomes important and needs to be evaluated when the exit velocity and/or temperature are sufficiently higher than ambient. Generally, this problem is solved by calculating an effective stack height (physical stack height plus a final plume rise height increment). ADPIC, however, can calculate the time- dependent plume rise of a stack release given the stack emission and meteorological parameters. The parameterizations developed for ADPIC by Desiato and Lange (1986) are used. These parameterizations rely primarily on the work of Briggs (1975) except for the unstable cases where the work of Weil and Houtt (1973) is used. Separate plume rise equations are used for the various combinations of momentum and buoyancy plume rise, vertical and bent-over plumes, and stability class.

The plume rise calculations are controlled with two groups of parameters input via the ADPIC.NML file. The first group are stack parameters for each source: plume rise flag for each source (USE\_PLUME\_RISE), internal radius of stack (STACK\_RADIUS), vertical exit velocity of gas (EXIT\_VEL), temperature of gas emitted from stack (STACK\_TEMP), and optional sensible heat emission rate (HEAT\_EMISSION\_RATE). The second group are time-varying meteorological parameters for each source: ambient air temperature (AMBIENT\_TEMP), height of strong inversion affecting plume rise (INV\_HGT\_AT\_STACK) and vertical temperature gradient (VERT\_TEMP\_GRAD).

#### **I.B.4.g. Explosive Cloud Rise**

An explosive cloud rise module has been incorporated into ADPIC which simulates additional forces acting upon the ADPIC particles due to their proximity to the thermal environment resulting from the detonation of a chemical explosive. When employed, this module generates additional vertical and horizontal components of motion applied to each particle remaining within the calculated effects of the rising thermal cloud.

A full description of the explosive cloud rise integral model code is found in Boughton and Delaurentis (1987), with further discussion of its use in ADPIC discussed in Foster, *et. al.* (1990). Briefly, the code provides a time evolution of the physical and thermodynamic properties of a buoyant cloud formed when a chemical explosive is detonated. The model is based on integrating the three dimensional conservation equations of mass, momentum, and energy over the cloud's cross section. With some simplifying assumptions the integral equations reduce to a set of ordinary differential equations which can be solved for the cloud radius, centerline height, temperature, and velocity as a function of time. These solutions are sensitive to the input parameters describing the ambient atmospheric conditions, especially the atmospheric temperature profile. The initial size and conditions of the cloud are determined using these atmospheric conditions and amount of explosive mass.

Once the cloud's characteristics are determined at each cloud rise model time step (which may be shorter than ADPIC's advection time step), the influence of this buoyant volume must be coupled to the ADPIC particles. There are two considerations in coupling the marker particles to the cloud, the initial loading of the particles into the calculational grid and their subsequent motion while they remain within the buoyant environment. All of the marker particles are initially loaded into the spherical cloud volume at the specified source release time. They have a Gaussian distribution within this volume with radial cutoff values equal to the initial cloud radius as determined by the cloud model. The standard deviations of this distribution are scaled to the cloud radius and are controlled through a scaling factor (SIG\_OVER\_R used to adjust the “flatness” of the initial distribution. For surface explosions, this initial sphere is tangent to the surface (e.g. it is centered at a height equal to the cloud radius).

As the cloud rises with time it is assumed to contain an environment of enhanced turbulence with entrainment occurring along its surface. Particles remaining inside the spherical cloud volume are treated differently than those moving beyond the cloud radius. Specifically, diffusion velocities are neglected while particles remain within the buoyant cloud. In addition, a fraction of the cloud's vertical motion is applied to each particle within radius,  $R$ , of the cloud center. The vertical motion imparted to each particle due to its proximity to the rising cloud is a function of its distance,  $r$ , from the cloud center, and is given by:

$$w = W \exp\left(-c(r/R)^2\right)$$

where

- $w$  = the particle's vertical lift due to the rising cloud (in addition to the vertical advection velocity),
- $W$  = the cloud's vertical lift motion,
- $c$  = a velocity coupling coefficient (= 0.35),
- $r$  = the particle's distance from cloud center, and
- $R$  = the cloud's radius.

Another difference in the treatment of particles within the cloud is a radially outward adjustment to maintain the particle's relative position between the cloud center and cloud edge as the cloud increases in size. This adjustment “velocity” simulates the effects of a well-mixed cloud, as it grows through entrainment, by redistributing the particles throughout the entire cloud volume. This velocity due to the cloud's expansion is independent from the particle's movement relative to the cloud center due to gravitational fall, the particle's vertical motion  $w$ , and potential differences in advection of the cloud and particles due to the spatial variation in the winds.

Therefore, particles remaining within the extent of the cloud are moved by applying the appropriate advection, expansion, fall, and vertical lift velocities. Once a particle leaves the influence of the rising thermal (usually due to the particle's gravitational fall velocity moving the particle beyond the cloud radius), only the usual advection, diffusion, and fall velocities are applied.

The dynamics of the cloud are followed until its vertical velocity decreases below a predetermined limit. At this point all particles are treated as though the cloud no longer exists and

the cloud calculation is bypassed for the remainder of the problem. A similar effect occurs if the cloud center rises above the top of the computational grid.

The use of the explosive cloud rise module is controlled by the ADPIC.NML parameter USE\_EXPL\_CLD\_RISE; the remaining explosive cloud rise parameters are contained in the ADPIC\_EXPL\_CLD namelist in the ADPIC.NML file. Although multiple ADPIC sources may be put into a single explosive cloud (controlled by parameter SRC\_EXPL\_CLD), the capability for multiple explosive clouds does not yet exist. The model used in ADPIC assumes a high explosive device (HE). Other explosive cloud source parameters include heat of detonation (HEATDET), high explosive amount (HEAMT), velocity coupling coefficient (VEL\_COUPL), and height of the center of the explosive cloud (ZINIT). Meteorological parameters required for the explosive cloud rise module include relative humidity at the surface (REL\_HUM), atmospheric pressure at the source center height (SRC\_PRESS), and the upper air temperature profile (TEMP\_PROFILE). The content of the output text file is controlled by the parameters PRINT\_EXPL\_CLD and PRINT\_TYPE\_EXPL\_CLD.

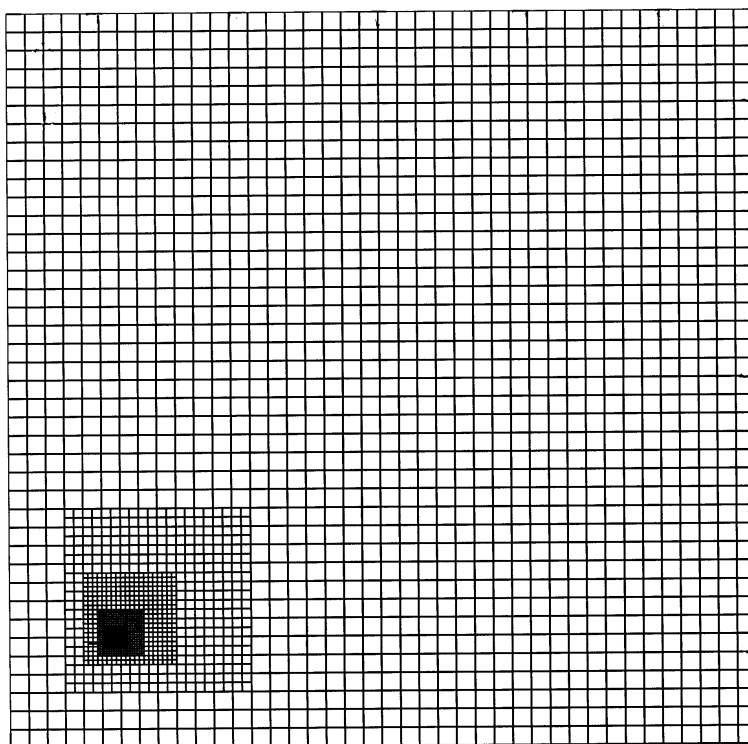
#### **II.B.4.h. Meteorological Parameters**

Several time-varying meteorological or meteorological-related parameters that affect the diffusion and/or deposition of pollutant sources in ADPIC can be input via the ADPIC.NML namelist file. These time-varying met parameters (TURB\_PARAM\_TYPE, STAB\_CLASS, INV\_OBUKHOV\_LEN\_ML, INV\_OBUKHOV\_LEN\_SL, ML\_HGT, SIGMA\_THETA\_FACT, and PRECIP\_SCAVENG\_COEF) may have up to 50 values, with the MET\_START\_DATE and MET\_START\_TIME parameters controlling the start date and time that each value will become effective.

#### **II.B.4.i. Nested Grid Sampling**

ADPIC output is in the form of mass (or activity) concentrations, with coordinates of the Lagrangian particles representing the pollutant mass and the advection, diffusion, and friction velocities. If the advection and sampling grid cells have the same resolution, the concentrations near the source area are chronically low because of an excessively high volume of dilution. (Concentrations are calculated by summing the mass contributions in each cell and dividing the result by the cell volume at specific times.) This problem corrects itself when the horizontal spread of the pollutant reaches the horizontal dimension of the sampling grid cell. A system of nested grids has been devised to improve the near-source resolution of concentration. The largest grid in Fig. II.B.8. represents ADPIC's standard advection and sampling grids, dimensioned 40 x 40 cells. All interior grids represent nested grids of diminishing size, each having 20 x 20 sampling cells at half the resolution of the preceding grid. In all cases, the vertical dimensions of the advection and sampling cells are identical. The source maintains the same relative position in each of the five grids. To demonstrate how these grids are situated in relation to one another, assume that the Model x coordinate and northing coordinates of the source are 410.5 km and 4011.3 km, respectively, and

that the assessor selects an advection grid for ADPIC whose lower-left coordinates are 400 km and 4000 km in the easting and northing directions. (Skewing the grid in this fashion would be appropriate when the prevailing winds and the winds in the foreseeable future are from the southwest.) Assuming a standard ADPIC grid having 40 x 40 cells in the horizontal with  $\Delta x$  and  $\Delta y$  equal to 2 km, the source would then occupy the advection cell whose index location is  $i = 6$  and  $j = 6$ . Since the nested grids have 20 x 20 cells, the source should reside on or within the  $i = 6$ ,  $j = 6$  cell of all nested grids. (The word “should” is used here since a nested grid, if necessary, is shifted to the east and south so that its boundary falls on the nearest mesh lines of the next largest grid.) After receiving this coordinate information from the TOPOG\*.GRID file (see Section III.C.1.), ADPIC correctly positions each of the four nested grids.



**Fig. II.B.8. Nested grids for improving near-source sampling**

ADPIC produces a set of five two-dimensional arrays of concentration at each level (the user can specify up to 10 levels). These arrays are used by the plotting code, PLCNT, to generate isopleths of concentrations. It should be clear from the previous discussion that several arrays can share a common space. When this occurs, PLCNT will always choose, in a spatial sense, the more highly resolved values of concentration. Use of the nested grid approach does, however, create complications; immediately beyond the extremity of any nested grid, the cell volume increases by a factor of four. This causes discontinuities in a contour when it spans two or more nested grids because of an abrupt drop in concentration. To improve the presentation, a routine has been developed that smooths the concentrations at the interface of two nested grids. However, use of this routine places a constraint on the ADPIC grid definition: the source must never be within two cells of

the advection grid boundary. Returning to our example, a UTM source location of 402 km east and 4010 km north would violate the two-cell condition, assuming that all grid selections remained the same.

There are four nested grids that are used along with each of these arrays. Each of the nested grids has a  $\Delta x$  and  $\Delta y$  one-half that of its next larger neighbor. These nested grids are only used for sampling concentrations. The standard computational grid is the only one used for advection and diffusion. These nested grids are fixed in that there are always four of them and their cell size is determined by the size of the computational cells.

The placement of the nested grid is determined by the placement of the source in the computational grid. The nested grid is automatically placed such that the source is in the same relation to the nested grid as the source has with the computational grid. The only restriction is that when the grid is originally set up, the source must be at least two grid cells away from the computational boundary. If the source has been placed too close to the outside, a warning message will be printed out in ADPIC with the option to continue or start over.

These concentration arrays are used by PLCNT to plot contours of the concentrations. In addition to these concentrations, ADPIC can also print out such things as particle coordinates, advection and diffusion velocities, cell concentrations, and friction velocities.

#### **II.B.4.j. Output**

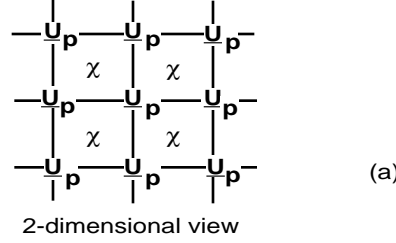
The contents of up to MAXSBINS (currently 10) ADPIC output bins may be specified using the parameters SOURCES\_TO\_SAMPLING\_BIN, SAMPLING\_TYPE and SAMPLING\_HGT. Any particular ADPIC source may be entered more than once. Also, combinations of ADPIC source numbers may be entered to sum the contributions from more than one ADPIC source.

#### **II.B.5. Accuracy of ADPIC**

Because the ADPIC time cycle is part Lagrangian and the original particle placement is determined by random number generator, the discussion of the accuracy of ADPIC must remain largely empirical.

##### **II.B.5.a. Finite Difference Algorithms**

As mentioned earlier, the cell concentrations are defined at the cell centers and the pseudo velocities  $\underline{U}_p$  are defined at cell corners. The pseudo velocities contain both the advection velocities  $\underline{U}$  and the diffusion velocities  $\underline{U}_D$ . Figure II.B.9 shows the diffusion velocity algorithm resulting from this arrangement, displayed in two dimensions for simplicity.



2-dimensional case

$$U_{x_D} = - \frac{K_{xx}}{\chi} \frac{\Delta \chi_x}{\Delta X}$$

$$U_{x_D} = - 2 \frac{K_{xx}}{\Delta X} \frac{(\chi_2 + \chi_3 - \chi_1 - \chi_4)}{(\chi_1 + \chi_2 + \chi_3 + \chi_4)}$$

$$0 < |U_{x_D}| < 2 \frac{K_{xx}}{\Delta X}$$

**Fig. II.B.9 Velocity algorithm: (a) grid mesh: (b) diffusivity.**

The truncation errors contained in the ADPIC diffusion velocity algorithm can be investigated by expanding it in a Taylor series. Consider the one-dimensional case.

$$U_{i+1/2} = - \frac{K_{i+1/2}}{\Delta X} \frac{(\chi_{i+1} - \chi_i)}{\chi_{i+1/2}}, \quad (\text{II.B.23})$$

Let the subscript be dropped for  $i + 1/2$  and expand  $\chi_{i+1}$  and  $\chi_i$  in Taylor series about  $i + 1/2$  to the first contributing error term:

$$U = - \frac{K}{\Delta X \cdot \chi} \left[ \chi + \chi' \frac{\Delta X}{2} + \frac{1}{2} \chi'' \left( \frac{\Delta X}{2} \right)^2 + \frac{1}{6} \chi''' \left( \frac{\Delta X}{2} \right)^3 - \left\{ \chi - \chi' \frac{\Delta X}{2} + \frac{1}{2} \chi'' \left( \frac{\Delta X}{2} \right)^2 - \frac{1}{6} \chi''' \left( \frac{\Delta X}{2} \right)^3 \right\} \right]; \quad (\text{II.B.24})$$

$$U = - \frac{K}{\Delta X \cdot \chi} \left[ \chi' \Delta X + \frac{1}{24} \chi''' \Delta X^3 \right] = - \frac{K \chi'}{\chi} - \frac{K}{24} \frac{\chi'''}{\chi} \Delta X^2. \quad (\text{II.B.25})$$

The first term on the right-hand side is the exact differential expression for the diffusivity velocity; the second term is the first contributing error term. Forming the ratio of the ADPIC diffusion velocity, Eq. (II.B.24), to the exact differential diffusion velocity, we obtain a measure of the importance of the error term

$$\frac{U_{\text{ADPIC}}}{U_{\text{EXACT}}} = 1 + \frac{\chi'''}{\chi'} \frac{(\Delta X)^2}{24}. \quad (\text{II.B.26})$$

#### II.B.5.b. The Number of Eulerian Cells Per Problem

It is important to have enough grid cells to resolve a given particle distribution so as to make the concentration gradients calculated from the cell concentrations representative of the actual gradients. While the particle positions per cell may indicate some subgrid resolution, it is important

to point out that the numerical solutions of the diffusion equation are defined on the Eulerian grid. To get a feeling for how well a distribution must be resolved to give reasonable answers, consider Eq. (II.B.26) above for the case of a one-dimensional Gaussian distribution:

$$\chi = \frac{Q}{\sigma} e^{-X^2/2\sigma^2} . \quad (\text{II.B.27})$$

The derivatives  $\chi'$  and  $\chi'''$  are

$$\chi' = -\frac{QX}{\sigma^3} e^{-X^2/2\sigma^2} . \quad (\text{II.B.28})$$

$$\chi''' = \frac{QX}{\sigma^5} \left( 3 - \frac{X^2}{\sigma^2} \right) e^{-X^2/2\sigma^2} . \quad (\text{II.B.29})$$

Substituting Eqs. (II.B.28) and (II.B.29) into (II.B.26),

$$\frac{U_{\text{ADPIC}}}{U_{\text{EXACT}}} = 1 + \frac{1}{24} \left( \frac{X^2}{\sigma^2} - 3 \right) \frac{(\Delta X)^2}{\sigma^2} . \quad (\text{II.B.30})$$

Equation (II.B.30) indicates the ADPIC and exact velocities are equal for  $x = \pm\sqrt{3}\sigma$ . (In this case,  $\chi''' = 0$ , and higher-order derivatives in the Taylor expansion of  $\chi$  must be retained to get a meaningful expression.) For  $X$  within  $x = \pm\sqrt{3}\sigma$ , ADPIC diffuses too slowly. For example, with  $x = \sigma$ ,

$$\frac{U_{\text{ADPIC}}}{U_{\text{EXACT}}} = 1 + \frac{1}{24} (-2) \frac{(\Delta X)^2}{\sigma^2} = 1 - \frac{1}{12} \frac{(\Delta X)^2}{\sigma^2} \quad (\text{II.B.31})$$

Similarly, when  $X$  lies outside  $\pm\sqrt{3}\sigma$ , ADPIC diffuses too quickly. Since 92% of all particles in a one-dimensional Gaussian distribution lie within  $X = \pm\sqrt{3}\sigma$ , the overall result is that ADPIC diffuses too slowly. Equations (II.B.31) show that the difference between the ADPIC and exact velocities, for any value of  $X$ , increases when the cell size  $\Delta X$  is large compared to the standard deviation of the particle distribution.

### II.B.5.c. The Number of Particles Per Cell

Inasmuch as the ADPIC particles represent a quantized density, it is desirable to have as many particles per cell as possible. The upper limit of this number is obviously dictated by computer storage. ADPIC assigns to each particle a fictitious cell volume of the same size and shape as the basic grid cell. Thereby, in calculating the concentration field from the particles, each particle will contribute some fractional part of its mass to eight nearest-neighbor cells in proportion to the overlap of its “volume” with the cell volumes of the neighbor cells. This procedure smooths out the concentration field and, in turn, the diffusion velocity field which is computed from it. This is consistent with the fact the Eulerian part of the cycle cannot resolve sub-grid-scale phenomena. From many prior calculations, we find that, due to this smoothing technique as few as one particle per cell on the average will yield meaningful results. With fewer particles per cell than that, i.e., when particles have no neighbors within one cell length around them, the ADPIC diffusivity velocity

algorithm (see Fig. II.B.2.) moves the particle to a grid point and “freezes” it there in the absence of any advection.

#### II.B.5.d. Time-step Calculation

ADPIC stability is rather insensitive to the length of the time step per cycle; however, since the  $i$ th particle transport is done by a forward differencing scheme,

$$\underline{X}_i^{n+1} = \underline{X}_i^n + \underline{U}_p^n \times \Delta t_i, \quad (\text{II.B.32})$$

diffusion occurs too fast when  $\Delta t_i$  becomes too large. The most satisfactory arrangement is to limit the transport of the fastest particle in the problem to one cell per time step. However, this rule can be relaxed in many cases. To calculate the time step, ADPIC finds the largest total velocity  $\underline{U}_p = \underline{U} + \underline{U}_D$  that any particle sees and computes a mean time step  $\Delta t$  according to the following equation:

$$\Delta t = F \cdot \text{Minimum of} \left[ \frac{\Delta X}{|U_{PM}|}, \frac{\Delta Y}{|V_{PM}|}, \frac{\Delta Z}{|W_{PM}|} \right], \quad (\text{II.B.33})$$

where  $F$  is an input number between 0 and 1.  $\Delta X, \Delta Y$ , and  $\Delta Z$  are the cell lengths and  $|U_{PM}|, |V_{PM}|$ , and  $|W_{PM}|$  are the absolute values of the maximum pseudo-velocities at any cycle. The individual particle time step ( $\Delta t_i$ ) is then generated by calling a random number generator, which generates  $\Delta t_i$  subject to the constraint  $0 \leq \Delta t_i \leq 2\Delta t$ .

#### II.B.5.e. Sub-grid Scale Diffusion

As shown in the section entitled “Accuracy of ADPIC,” when the particle distribution in ADPIC cannot be resolved by the grid mesh, errors in diffusion rates occur. This situation is found when one tries to model sources that are initially considerably smaller than a cell volume. Under such conditions, an assumption is made about the shape of the source particle distribution at early times until the distribution has spread out enough to represent an extended source, resolved by the grid mesh. The source particles can be diffused in accordance with such a distribution until such a time. The sole purpose of this source particle treatment is to provide ADPIC with a reasonably representative extended source for the study of larger-scale three-dimensional phenomena involving varying meteorology, diffusion parameters and topography over distances at which the effect of the source is not important. If effects near the source are to be investigated, the close-in ADPIC problem will have to be run separately. With this in mind, it is assumed the sub-grid scale source is Gaussian in character and that it remains locally Gaussian until its standard deviation  $\sigma$  is approximately equal to the size of a grid cell and represents an extended source. At this point, the locally Gaussian assumption is discarded and ADPIC calculates diffusion by its concentration gradient method. It will now be shown how this assumption leads to a subgrid scale diffusivity velocity, and how this velocity is integrated to diffuse source particles. This approach treats horizontal and vertical diffusion of the particles independently. For simplicity, the case for the cross-wind ( $y$ ) direction is illustrated.



For a Gaussian distribution, the concentration field for a puff is described by

$$\chi = \frac{Q}{(2\pi)^{1/2} \sigma_y} \exp - \left( \frac{1}{2} \frac{y_D^2}{\sigma_y^2} \right) \quad (\text{II.B.34})$$

where  $\chi$  is the particle concentration,  $Q$  is the source,  $\sigma_y$  is the standard deviation and  $y_D$  is the distance of the particle from the center of the distribution. By dividing the diffusion flux

$$\chi V_D = -K_y \frac{\partial \chi}{\partial y} \quad (\text{II.B.35})$$

through by  $\chi$ , the definition of diffusion velocity is obtained

$$V_D = -\frac{K_y \partial \chi}{\chi \partial y} = -K_y \frac{\partial(\ell n \chi)}{\partial y}, \quad (\text{II.B.36})$$

where  $V_D$  is an average diffusion velocity and  $K_y$  the diffusion parameter. Calculating  $\ell n(\chi)$  from Eq. (II.B.34) and substituting in (II.B.36), the diffusivity velocity becomes

$$V_D = -K_y \frac{\partial}{\partial y} \left[ \ell n \frac{Q}{(2\pi)^{1/2}} - \ell n \sigma_y - \frac{1}{2} \frac{y_D^2}{\sigma_y^2} \right] = \frac{K_y y_D}{\sigma_y^2}. \quad (\text{II.B.37})$$

Substituting Eqs. (II.B.9) and (II.B.11) into the equation for  $V_D$  ( $C$  has been set equal to 1), Eq. (II.B.37) yields

$$V_D = \frac{y_D}{\left( \frac{\sigma_0^2}{\varepsilon} \right)^{1/3} + \frac{2}{3} t}. \quad (\text{II.B.38})$$

This diffusivity velocity is independent of the ADPIC grid mesh and is defined at the coordinate  $(y_d, t)$  of a given source particle. [The quantity  $y_D$  is the absolute value of the distance from the particle to the diffusion (puff) center, as can be seen from its definition in Eq. (II.B.34).] The variable  $t$  represents the “age” of the particle since its creation and coincides with the actual ADPIC problem time only in the case of a single puff source.

Equation (II.B.38) shows that in the case of a true point source, when  $\sigma_0 \rightarrow 0$ , or at a late time when  $\left( \sigma_0^2 \right)^{1/3} \ll \frac{2}{3} t$  for any size source,

$$V_D = \frac{y_D}{\frac{2}{3} t}. \quad (\text{II.B.39})$$

Hence it is a property of Gaussian diffusion that all those particles, for which the diffused distance  $y_D$  divided by the diffusion time  $t$  is a constant ratio, will have the same diffusion velocity  $V_D$ . Also, Eq. (II.B.38) shows that if a particle gets generated as part of a Gaussian source distribution of size  $\sigma_0$  at a distance  $y_0$  at  $t = 0$ , its initial diffusion velocity is

$$V_D = \frac{y_0}{\left( \sigma_0^2 / \varepsilon \right)^{1/3}}. \quad (\text{II.B.40})$$

For the case of constant  $K$ , Eqs. (II.B.9) and (II.B.11) become, respectively,

$$K = K_{\text{constant}} \quad (\text{II.B.41})$$

$$\sigma_y^2 = \sigma_0^2 + 2 K t . \quad (\text{II.B.42})$$

A derivation similar to the scale-dependent case leads to an expression for the diffusivity velocity

$$V_D = \frac{y_D}{\frac{\sigma_0^2}{K} + 2 t} . \quad (\text{II.B.43})$$

Since the diffusivity velocity can also be expressed as

$$V_D = \frac{\partial y_D}{\partial t} . \quad (\text{II.B.44})$$

Eq. (II.B.43) can be integrated over time as long as neither  $\sigma_0$  nor  $K$  are assumed strong functions of  $y$  or  $t$ . This integration is carried out each ADPIC cycle over the applicable time step  $\Delta t$ . From Eqs. (II.B.43) and (II.B.44) for the constant  $K$  case, the integral equation is

$$\int_{y_D}^{y_{D+\Delta y}} \frac{dy_D}{y_D} = \int_t^{t+\Delta t} \frac{dt}{\frac{\sigma_0^2}{K} + 2 t} . \quad (\text{II.B.45})$$

Integrating, and solving for  $\Delta y_D$ ,

$$\Delta y_D = y_D \left\{ \left[ 1 + \frac{\Delta t}{\left( \sigma_0^2 / 2 K \right) + t} \right]^{1/2} - 1 \right\} \quad (\text{II.B.46})$$

where  $\Delta y_D$  is the distance diffused by the source particle during time step  $\Delta t$ ,  $y_D$  is the diffused distance at the beginning of the cycle,  $\Delta t$  is the ADPIC time step and  $t$  is the “age” of the source particle at the beginning of the cycle.

A similar integration of Eq. (II.B.38) for scale-dependent diffusion results in

$$\Delta y_D = y_D \left\{ \left[ 1 + \frac{\Delta t}{\frac{3}{2} \left( \frac{\sigma_0^2}{\varepsilon} \right)^{1/3} + t} \right]^{3/2} - 1 \right\} . \quad (\text{II.B.47})$$

The actual distance traveled by a source particle during one cycle in the ADPIC grid can now be computed by adding the distance  $\Delta y_A$  from advection to the distance  $\Delta y_D$  diffused during a time step:

$$\Delta y = \Delta y_D + \Delta y_A = \Delta y_D + V_A \cdot \Delta t \quad (\text{II.B.48})$$

where  $V_A$  is the interpolated advection velocity obtained in the same way as for regular non-source ADPIC particles.

The various updated values at cycle  $n + 1$ , the actual source particle coordinate  $y$ , the diffusion distance  $y_D$ , and the particle age  $t$  are then calculated and stored for each individual source particle for the next cycle, i.e.,

$$y^{n+1} = y^n + \Delta y \quad (\text{II.B.49})$$

$$y_D^{n+1} = y_D^n + \Delta y_D \quad (\text{II.B.50})$$

$$t^{n+1} = t^n + \Delta t \quad (\text{II.B.51})$$

For continuous sources, newly-generated puffs are composed of source particles having Gaussian distributions of size  $\sigma_0$ . All particles generated during the same cycle will have a variety of initial diffusion distances  $y_D$  according to the Gaussian distribution, but all will have the same initial age, namely  $t = 0$ . After one cycle of diffusion in the manner discussed above, their age will be  $t = \Delta t_1$ , where  $\Delta t_1$  is the particle diffusion time for its first cycle, and after two cycles  $t = \Delta t_1 + \Delta t_2$  and so on. So at any time there will exist several puffs of source particles, each of different age. When a given puff of source particles that started out with a standard deviation  $\sigma_0$  representative of the source size has diffused to a distribution size with a standard deviation  $\sigma \simeq \Delta Y$ , where  $\Delta Y$  is the ADPIC cell length, the whole set is considered a suitable extended source. It is therefore released from its source particle status and is considered to consist of regular ADPIC particles. The test for release is made against  $t$ , the age of the particle set, at every time step. In the meantime, source particles are subject to the same boundary conditions as regular particles. While the source particles are treated independent of regular ADPIC particles as a sub-grid scale distribution, they are counted together with the regular particles in determining the particle concentration field in ADPIC, which is used to calculate the diffusion velocities for the regular particles.

## **II.C. Support Routines**

### **II.C.1. TOPOG**

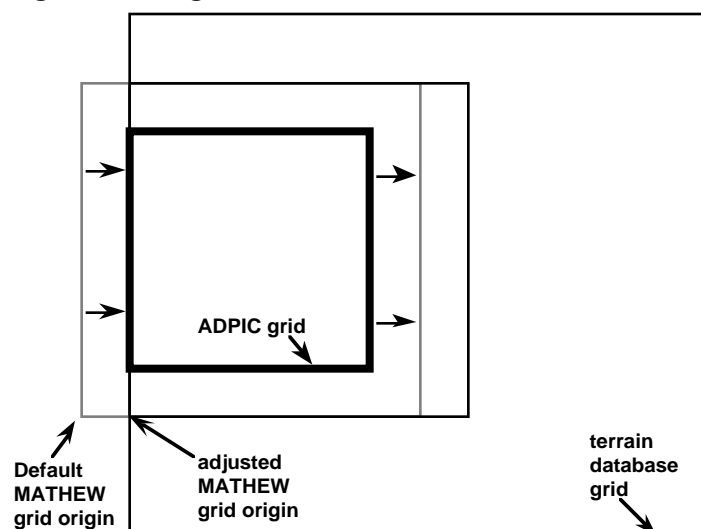
The construction of a solid bottom boundary for the CG-MATHEW and ADPIC models is accomplished through TOPOG. The boundary can be viewed as a system of blocks which approximate the shape of the actual terrain. Therefore the results of subsequent CG-MATHEW and ADPIC runs will reflect the major topographic features in the area of interest. The system of blocks is derived by averaging data from a regional elevation data base (see Section III.B.3.) to a grid volume defined by the user (see Section III.B.1.). Because of the large amount of averaging that has been applied to the original digitized terrain data in the generation of the system of blocks, the model terrain produced automatically by TOPOG may not be the best representation of terrain possible for a given grid resolution. ARAC's in-house version of TOPOG allows a user to selectively and interactively raise or lower individual boundary cells so as to more closely conform to the actual surface. Appendix C provides a description of this interactive capability.

The definitions of the CG-MATHEW and ADPIC grid volumes produced by TOPOG are made up of grid parameters and terrain heights for each horizontal grid cell. The user selects a grid by assigning values to certain grid parameters in the TOPOG.NML file (see Section III.B.1.). The ADPIC and CG-MATHEW grids are subject to a number of constraints in order to ensure proper averaging. TOPOG checks the grid parameters and if they define an illegal grid, TOPOG either halts with diagnostic errors or makes minor adjustments to some of the parameters in an attempt to create a legitimate grid.

In a typical ARAC problem, the user specifies the center of the ADPIC grid in the  $x$  and  $y$  coordinates of the map projection selected for the problem (optionally the user can specify the coordinates of the lower left corner of the ADPIC grid). Currently, most regional problems are run on the Universal Transverse Mercator (UTM) projection. The user also specifies the  $x$ ,  $y$  and  $z$  ranges of the grid (or optionally the  $x$ ,  $y$ , and  $z$  dimensions of an individual cell). Since the number of grid points in each dimension is normally left constant, choosing the cell size is equivalent to specifying the range of the ADPIC grid (e.g., an ADPIC grid having 41 by 41 by 15 points with a cell size of 1 km by 1 km by 50 m defines a grid volume of 40 km by 40 km by 700 m). The CG-MATHEW grid is defined on the basis of the user's ADPIC grid. They have the same grid cell sizes, but the number of points in CG-MATHEW's grid system (in any direction) must, at a minimum, be as great as the number in ADPIC's grid system. The CG-MATHEW grid surrounds the ADPIC grid so that a band of cells borders on all sides of the ADPIC grid. This has a buffering effect that protects ADPIC from possibly poor adjustments of the winds near CG-MATHEW's boundaries. For the standard ARAC model grid dimensions (CG-MATHEW: 51 by 51 by 15 grid points, ADPIC: 41 by 41 by 15 grid points), the band surrounding the ADPIC grid is usually five grid cells wide.

The terrain averaging requires that the boundaries and origins of both the CG-MATHEW and ADPIC grids lie on top of grid points in the elevation data base grid. The current averaging procedure

also requires that both of the horizontal grid cell lengths (DELX, DELY) be integer multiples of one-quarter the resolution of the terrain data base. Since standard regional elevation grids have grid cells which are 500m by 500m, legitimate values for DELX and DELY are integer multiples of 125 m. For hemispheric problems the grid cells must be 381 km by 381 km. The constraints on the cell size are absolute. TOPOG will halt if illegal values for DELX and DELY are specified; however, TOPOG will shift the CG-MATHEW and ADPIC grids as much as DELX (or DELY, as appropriate) to ensure the overlaying of the proper points. The origins will be shifted as little as possible from the positions given in TOPOG.NML. The other constraint on the grids is that no part of the CG-MATHEW grid may lie outside of the terrain database grid, and that no part of the ADPIC grid may lie outside of the CG-MATHEW grid. If the ADPIC grid, as specified, lies within the terrain data base grid and the default CG-MATHEW grid does not, then TOPOG will attempt to shift the CG-MATHEW grid to create a legal system of grids (See Fig.II.C.1.).



**Fig. II.C.1. Example of the automatic shift of the CG-MATHEW grid by TOPOG to create a permissible system of grids.**

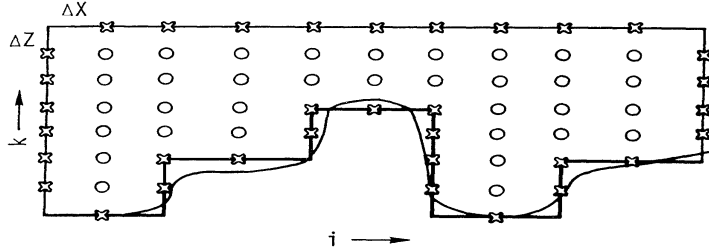
The ADPIC grid can be specified as being as much as one DELX or DELY step out of the area covered by the regional elevation file grid. If so, the ADPIC grid will be shifted so that the appropriate boundary matches the elevation file grid boundary. The CG-MATHEW grid will also be shifted as described above to match the elevation file grid boundary along the corresponding border. If the ADPIC grid lies further from the border that one CG-MATHEW/ADPIC grid step, then an error is reported and TOPOG ends.

If there is no important terrain relief in an area or if no elevation data is available, TOPOG can generate a flat grid of arbitrary size (see FLAT\_TOPO in Section III.B.1.). There are then no constraints on the CG-MATHEW and ADPIC grids with respect to the elevation data base grid. Regardless of the means of selecting a grid (or even if errors are discovered during validation), all grid parameters are displayed on the terminal and in the TOPOG.LOG file to allow the user to determine the acceptability of the model grids produced by TOPOG, or to determine the cause of any errors.

The final CG-MATHEW grid, along with the vertical grid spacing (DELZ) selected by the user, defines a system of grid cells filling a grid volume. Terrain is represented by specifying which grid cells in a vertical column are open (wind is permitted to pass through at least one of the cell's faces) and which cells are closed (i.e., underground). The terrain is described by a two-dimensional array of integers with each element in the array specifying the integer number of grid cells that lie underground in the CG-MATHEW grid.

TOPOG generates the terrain arrays by averaging the data in the regional elevation data base over the CG-MATHEW grid. The standard elevation data file specifies elevations for a region covered by a grid of square cells oriented along the appropriate map coordinate system (typically UTM coordinates for regional problems). Elevation data base grids are dimensioned 400 by 400 with each grid cell commonly a 500 m square (in such cases the area described is 200 km by 200 km). The terrain file contains an array of values with each value being the average height, in meters above sea level, of the corresponding grid cell. Since the grid has been accepted by TOPOG before the averaging begins, each CG-MATHEW cell is an integer multiple of one quarter the elevation grid spacing and, as a result, the average height of a CG-MATHEW cell can be found by averaging the heights of the portions of the elevation data base grid cells it contains with the averaging weights reflecting the area of the portion of the elevation grid cell contained in the CG-MATHEW grid cell. On completing this step a height in meters above sea level is associated with each CG-MATHEW cell. In the past, the terrain averaging done in TOPOG was based on CG-MATHEW double cells, i.e., four CG-MATHEW grid cell in a 2 by 2 array were averaged as an entity. This was used to avoid single cell holes in the model terrain that permit ADPIC particles to diffuse into them but cannot contain any winds, since five of the six faces are closed, so such particles cannot be advected out of them. Current procedures in TOPOG, largely interactive, allow single cell holes to be removed in other ways (automatically if the code is being run non-interactively) so this option is rarely used in ARAC applications. However, TOPOG is occasionally ported to other systems that may not be able to support the necessary graphics tools, so the option for this double cell averaging is maintained (see DOUBLECELL in Section III.B.1.),

To complete the averaging, each cell height is rounded to the nearest integer multiple of DELZ. The lowest value for all the cells in the grid is then subtracted from each grid value, thereby re-defining the bottom of the CG-MATHEW grid volume. The integer height of each cell is then assigned to the appropriate CG-MATHEW grid cell. This array is written to the TOPOG\*.GRID file for use by the remaining models. The ADPIC cell heights are extracted directly from this array when ADPIC reads the TOPOG\*.GRID file, accounting for any offsets due to the typically larger CG-MATHEW grid dimensions. CG-MATHEW requires its terrain heights to be offset so that the bottom of the grid has a value of one. Hence, both MEDIC and CG-MATHEW increment each cell height by one when they read the TOPOG\*.GRID file. A two-dimensional illustration of the terrain representation is shown in Fig. II.C.2.



**Fig. II.C.2. Illustration of the topography specification in two dimensions.**

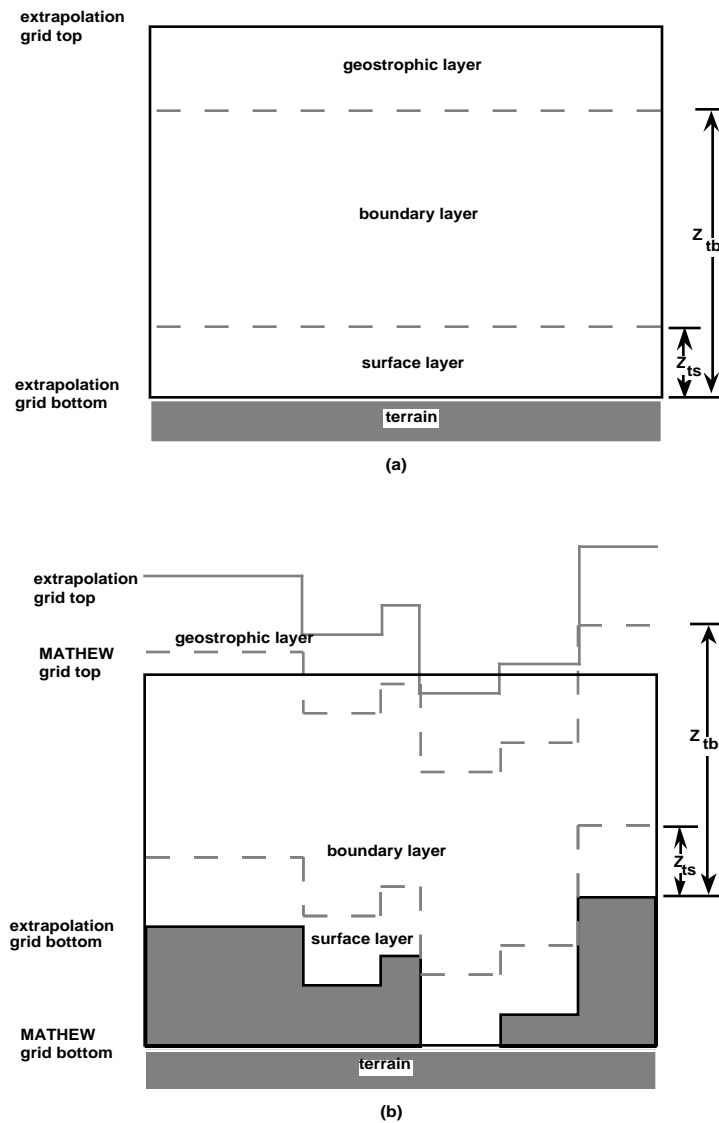
## II.C.2. MEDIC

Since CG-MATHEW adjusts the extrapolated winds by the *minimum* necessary amount in producing a mass-consistent arrangement, a reasonable, first-guess field of wind velocities,  $\underline{U}^0$  in Eq. (II.A.1), must be supplied by MEDIC, the wind extrapolation code. Attention to detail in the preparation of the extrapolated field is critical. Fairly minor variations in the parameters input to MEDIC can lead to grossly different wind fields, all of which are consistent with the measured data. For this reason, we rely heavily on the interpretive ability of an experienced meteorologist to determine the acceptability of a result. Using the tools of his trade (e.g., weather maps, local observations, upper air data, pilot reports), a meteorologist can factor in all of the known and relational information (in the form of model input) to produce a realistic and meteorologically consistent wind field. If necessary, the model is rerun with altered input until the desired result is achieved. This procedure is extremely important in that physical processes, many of which are not modeled in CG-MATHEW, can be implicitly represented in the structure of the extrapolated field.

The creation of an extrapolated gridded wind field from available wind observations is accomplished by one of two related algorithms. The first method uses surface observations and user-defined parameters to construct the required three-dimensional wind field with minimal upper air data required. This method is referred to as parameterized extrapolation; it is used when no upper air soundings are available or when the credibility of a sounding is in doubt. The other approach makes use of all surface and upper air observations in the wind extrapolation process. This approach is called profile extrapolation; it is the preferred method if at least one local upper air sounding, taken within the past six hours, is available. Parameterized extrapolation is the default in MEDIC (see (PROFILE\_EXTRAP in Section II.B.4.).

Three separate layers of the atmosphere are recognized in the extrapolation procedures. The surface layer is the lowest layer in which surface effects predominate and no direction shear with height is allowed. The highest layer, the geostrophic layer, is the region of the atmosphere where surface effects have virtually dissipated and the winds reflect larger-scale synoptic flow. Wind speed and direction are assumed to be constant with height in this layer, i.e., there is no wind shear. The central layer, or boundary layer, is the region where conditions are intermediate between the surface and geostrophic layers; both speed and directional shear can occur.

Both extrapolation procedures produce two three-dimensional arrays of horizontal wind components, one array for the  $u$ -components and the other array for the  $v$ -components, containing the wind vectors at each grid point of a volume having the same dimensions as the CG-MATHEW grid. However, unlike the CG-MATHEW grid, the extrapolation grid volume is adjacent to but entirely above the model terrain surface (i.e., in the extrapolation grid, heights above grid bottom are always equal to heights above model terrain). On completion of the extrapolation, model terrain is pushed up through the grid bottom with a corresponding upward shift in the extrapolated winds. Thus, the atmospheric layers defined in the extrapolation process ( $Z_{sl}$  and  $Z_{bl}$ ) are flat in the extrapolation grid volume and have the shape of the terrain in the CG-MATHEW grid volume (see Fig. II.C.4.). This produces the wind fields that are passed to CG-MATHEW. CG-MATHEW transforms the winds into face-centered components for ease of calculation (see Section II.A.3.).



**Fig. II.C.4. Schematic of a CG-MATHEW grid in two dimensions: (a) before, and (b) after the introduction of model terrain.**



### II.C.2a. Parameterized Extrapolation

Parameterized extrapolation involves two basic steps. First, wind speed and direction values must be assigned to two height levels in each column of grid points of the extrapolation grid volume. One level is near the surface and the other level is at or above the top of the grid. Second, winds along the columns of grid points are determined by interpolating between the wind values at the two levels, and by extrapolating to the surface below the lower level. The interpolations and extrapolations are controlled by a number of user-defined parameters that aid in simulating the vertical structure of the atmosphere.

The near-surface level wind for each column of grid points is derived from surface wind observations. First, all surface speed measurements are normalized to a user-specified height above terrain, called the reference height (see REF\_HGT in Section III.B.5.). In order to minimize the alteration caused by normalization, REF\_HGT is usually chosen to be the height above terrain of most of the observing instruments. The surface layer power law formula used in the normalization is

$$S_{\text{norm}} = S_{\text{obs}} (Z_{\text{ref}} / Z_{\text{obs}})^{\gamma_{\text{sl}}} , \quad (\text{II.C.1})$$

where  $S_{\text{norm}}$  is the wind speed at height REF\_HGT,  $S_{\text{obs}}$  is the observed wind speed,  $Z_{\text{ref}}$  is the reference height,  $Z_{\text{obs}}$  is the observation height, and  $\gamma_{\text{sl}}$  is the power law exponent for the surface layer (see PWR\_LAW\_EXPNT\_SL in Section III.B.5. and the description of the OBSERV.MET file in Section III.B.7). Wind directions are not altered when normalization occurs since directions rarely show significant changes for typical adjustment heights. The reference level speed and direction for a column of grid points are then derived from the normalized measurements by an extrapolation based on the inverse square of the distance to nearby observation points according to the formula:

$$S_{\text{ref}} = \frac{\sum_{n=1}^N S_n / R_n^2}{\sum_{n=1}^N 1 / R_n^2} \quad (\text{II.C.2})$$

where  $S_{\text{ref}}$  is the reference level speed for the grid point column,  $N$  is the number of nearest observations being used in the extrapolation,  $S_n$  is the normalized speed measurement for a given station, and  $R_n$  is the distance from the horizontal location of the column to the station. The four nearest stations are normally used, although this can be changed by recompiling. The reference level direction for the column is calculated by separate inverse-square extrapolations of the sines and the cosines of the observed values based on formulas analogous to Eq. (II.C.2.). The sines and cosines are extrapolated separately to avoid the discontinuity caused by interpolating through  $0^\circ$  ( $360^\circ$ ). and are then used to generate a consistent angle using an arctangent function. The subset of the observation stations used to determine the direction may not be the same as the subset for speed since a station may report only speed or direction. Therefore the selection of the nearest observation stations is performed separately for the speed and direction extrapolations. At least one surface speed and one surface direction observation, not necessarily at the same point, must be input to allow the extrapolation to proceed.

The upper level wind in a grid point column is generated from upper air wind observations. The upper level is usually taken to be the top of the extrapolation grid volume. However, if the top of the boundary layer is above the top of the extrapolation grid, the wind is instead determined at the top of the boundary layer. In order to extract values at arbitrary heights from the discrete upper air profile measurements provided by a sounding, it is assumed that winds vary linearly between successive measurement heights. Directions vary through an arc describing the smallest angle between two measurements. Above the highest measurement level, winds are assumed to remain constant with height. Winds below the lowest measurement level are taken to be constant down to the top of the surface layer and to obey a power law for speed within the surface layer. The inverse square extrapolation procedure [see Eq. (II.C.2)] is then used to assign a speed and direction at the upper level of each grid point column. Profiles with speed measurements only or direction measurements only will be accepted as input. At least one upper air speed measurement and one upper level direction measurement, not necessarily at the same location, must be input to allow the extrapolation to proceed.

Having extrapolated winds at two levels, values can now be assigned to all the grid points in a column using an interpolation method that reflects the basic three layer structure of the atmosphere. For grid points in the surface layer above and below the reference height, the wind speed is determined using a power law formula similar to that used for the reference level normalization (Eq.II.C.1) :

$$S(Z) = S_{\text{ref}} \left( Z/Z_{\text{ref}} \right)^{\gamma_{\text{sl}}} \quad (\text{II.C.3})$$

where  $S$  is the speed at a surface layer grid point,  $Z$  is the height of the surface layer grid point, and  $S_{\text{ref}}$  and  $Z_{\text{ref}}$  are the reference level speed and height for the column of grid points. Since there is no wind shear in the surface layer, the reference level direction is assigned to all surface layer grid points of the column. The power law formula is also used to generate a surface slip velocity at the surface roughness height,  $Z_{\text{srh}}$  (see SFC\_ROUGH\_HGT in Section III.B.5.). In the geostrophic layer, constant winds are assumed and the upper level speed and direction are assigned to all grid points in the column above the boundary layer.

Winds in the boundary layer are computed so as to provide continuity between the surface layer winds and the geostrophic layer winds. The simplest way of satisfying this condition is to linearly interpolate between the winds at the top of the surface layer and the top of the boundary layer. This capability is provided and is adequate for many situations. However, atmospheric conditions can arise which require greater generality in the interpolation. The speed shear in the boundary layer is often variable, with the greatest change occurring near the top, as frequently occurs when the top of the boundary layer coincides with a temperature inversion. This behavior may be modeled by a power law interpolation formula with a user-defined exponent. Another important consideration in the boundary layer is the variation of direction with height. The difference between any two angles can be determined along two directions of rotation, clockwise or counter clockwise. As a result, there are two possible interpolations between the direction at the top of the surface layer and the direction

at the top of the boundary layer. In middle latitudes of the northern hemisphere, winds usually change in a clockwise direction with increasing height, i.e., they “veer” with height. Therefore, it is not always advisable to determine angular difference through the smallest angle. There are also cases where the wind directions turn counter clockwise with height, i.e., they “back.” This can occur, for example, when a front moves through an area. It is important for an assessor to be able to control the sense of the directional shear in the boundary layer, a capability which is provided in MEDIC.

Interpolation in the boundary layer is accomplished using the following formulas:

$$S(Z) = \frac{\Delta S_{bl} (Z - Z_{ts})^{\gamma_{bl}}}{(\Delta Z_{bl})^{\gamma_{bl}}} + S_{ts} , \quad (\text{II.C.4a})$$

$$\theta(Z) = \frac{\Delta' \theta_{bl} (Z - Z_{ts})^{\gamma_{bl}}}{(\Delta Z_{bl})^{\gamma_{bl}}} + \theta_{ts} \quad (\text{II.C.4b})$$

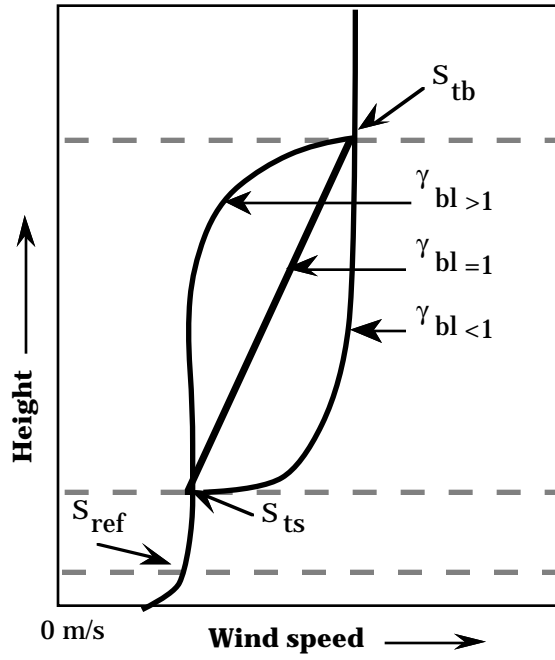
where  $\gamma_{bl}$  is the boundary layer extrapolation exponent (see EXTRAP\_EXPNT\_BL in Section III.B.5.).  $\Delta S_{bl}$  is the speed shear over the depth of the boundary layer, i.e.,  $\Delta S_{bl} = S_{tb} - S_{ts}$  where  $S_{tb}$  and  $S_{ts}$  are the speeds at the top of the boundary layer and the top of the surface layer, respectively.  $\Delta Z_{bl}$  is the depth of the boundary layer, i.e.,  $\Delta Z_{bl} = Z_{tb} - Z_{ts}$  where  $Z_{tb}$  and  $Z_{ts}$  are the top of the boundary layer and the top of the surface layer heights, respectively (see BL\_HGT and SL\_HGT in Section III.B.5.).  $\Delta' \theta_{bl}$  is the direction difference over the depth of the boundary layer. This difference depends on  $\theta_{tb}$  and  $\theta_{ts}$ , the directions at the top of the boundary layer and the surface layer, respectively, and  $\theta_r$  (see MAX\_VEER\_PARAM\_VERT in Section III.B.5.) according to the following recipe:

$$\Delta \theta = \theta_{tb} - \theta_{ts} . \quad (\text{II.C.5})$$

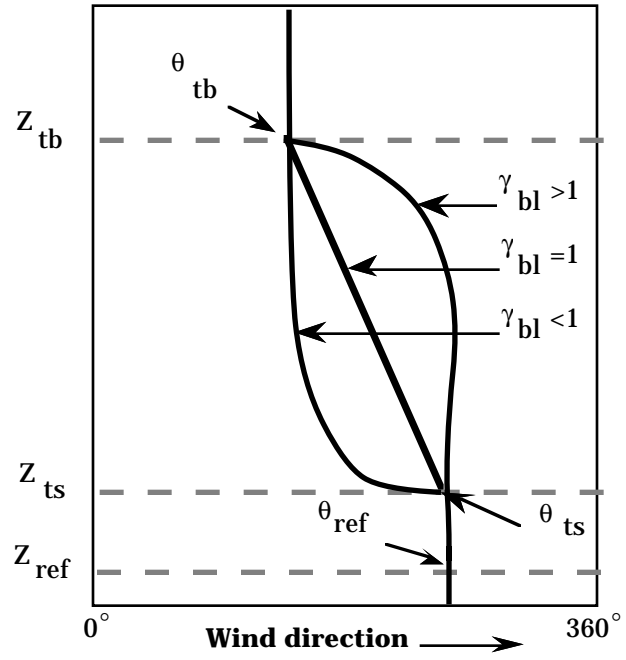
IF  $\theta_r - 360^\circ \leq \Delta \theta \leq \theta_r$ , then  $\Delta' \theta_{bl} = \Delta \theta$  ;  
 if  $\Delta \theta < \theta_r - 360^\circ$ , then  $\Delta' \theta_{bl} = \Delta \theta + 360^\circ$  ,  
 if  $\Delta \theta > \theta_r$ , then  $\Delta' \theta_{bl} = \Delta \theta - 360^\circ$  .

The assignment of a value to  $\theta_r$  causes wind directions to veer with height if the directional shear, determined in the clockwise sense, is less than  $\theta_r$  ; it causes wind directions to back with height if the clockwise shear is greater than  $\theta_r$ . Equations (II.C.4a) and (II.C.4b) are evaluated at the heights of all the grid points within the boundary layer. When the entire process described above is completed for every column of grid points in the grid, the extrapolation volume is ready to be examined by the assessor. The assessor can adjust the parameters in the extrapolation ( $Z_{srh}$ ,  $Z_{ref}$ ,  $Z_{ts}$ ,  $Z_{tb}$ ,  $\gamma_{sl}$ ,  $\gamma_{bl}$ , and  $\theta_r$ ) if the extrapolated field does not provide a reasonable picture of the atmosphere. The effects of two of the parameters,  $\gamma_{bl}$  and  $\theta_r$  are shown in Figs.II.C.5a-d.

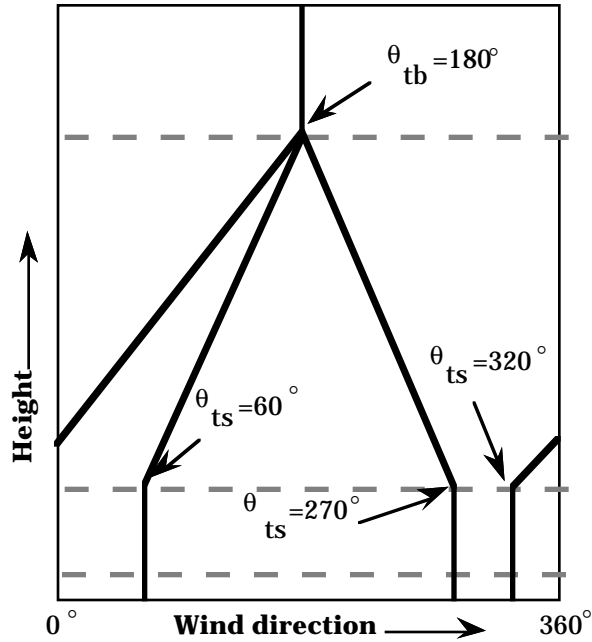
d.



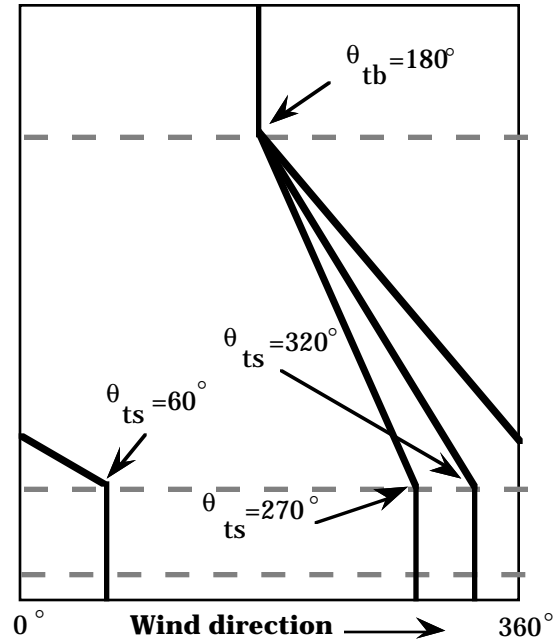
(a) Effect of  $\gamma_{bl}$  on speed in parameterized profile method



(b) Effect of  $\gamma_{bl}$  on direction in parameterized profile method



(c)



(d)

Fig. II.C.5. Effect of (c)  $\theta_r = 240^\circ$  and (d)  $\theta_r = 100^\circ$  on various values of  $\theta_{ts}$  in parameterized profile method.

As mentioned previously, the parameterized extrapolation method is normally used only in the absence of acceptable upper air data. When upper air data is unavailable for a site, an estimate of the upper level winds must be supplied. A reasonable evaluation of the wind speed and direction might be extracted from synoptic charts for an appropriate height. If no other basis can be found for a selection, climatological data might be helpful in providing an estimate. If a recent, nearby upper air sounding is available, then the profile extrapolation method should be used because this procedure takes greater advantage of the available information.

### **II.C.2b. Profile Extrapolation**

The profile extrapolation method provides a mechanism for the controlled melding of surface and upper air wind observations controlled by the same variables used in the parameterized method, although some of the variables are used in slightly different ways. It is assumed that surface observations provide the most reliable picture of the actual winds in the lower levels of the atmosphere and should be weighted more heavily than upper air observations in any interpolation scheme since the density of surface stations is almost always much higher than the density of upper air stations. For a typical vertical grid dimension, an upper air sounding provides the best wind information for extrapolation near the top of the grid, while at intermediate grid point levels, the relative importance of the surface and upper air observations will vary with the meteorological conditions. It is also assumed in this method that the “shape” of a measured profile, i.e., its general features, can be applied to different locations in a region, even if the actual speed and direction values at a given level do not match those for the other positions.

Three main steps are required in the profile extrapolation method. First, a wind speed and direction derived from the surface observations must be assigned to each column of grid points in the CG-MATHEW grid at a level near the bottom of the extrapolation grid. Next, each grid point in a column is assigned speed and direction values which are derived entirely from the upper air soundings, i.e., a profile is synthesized for the grid point column from the upper air data alone. Finally, the surface wind is combined with the synthesized profile to give an extrapolated wind at each grid point in the column. The surface and upper air components are weighted differently at different heights, with the weighting being controlled by various user-defined parameters.

The near surface wind at the reference height  $Z_{\text{ref}}$  is generated from surface observations using the same procedure as in parameterized extrapolation. For grid points below  $Z_{\text{ref}}$  the wind speed is found according to Eq. (II.C.3) and the slip velocity is found by applying Eq. (II.C.3) at the surface roughness height.

A synthesized profile is created along the column of grid points by inverse square extrapolation of the nearest upper air observations at the same height (Eq.II.C.2). It consists of wind speeds and directions for each grid point in the column  $S_{\text{syn}}(Z)$  and  $\theta_{\text{syn}}(Z)$ , as well as for the reference level  $Z_{\text{ref}}$ , the top of the surface layer ( $Z_{\text{ts}}$ ), and the top of the boundary layer ( $Z_{\text{tb}}$ ).

In general, the near-surface wind derived from the surface data will not match the winds derived from upper air data. The two are meshed by implementing a height dependent correction term which is added to the synthesized profile value at each grid point in the column:

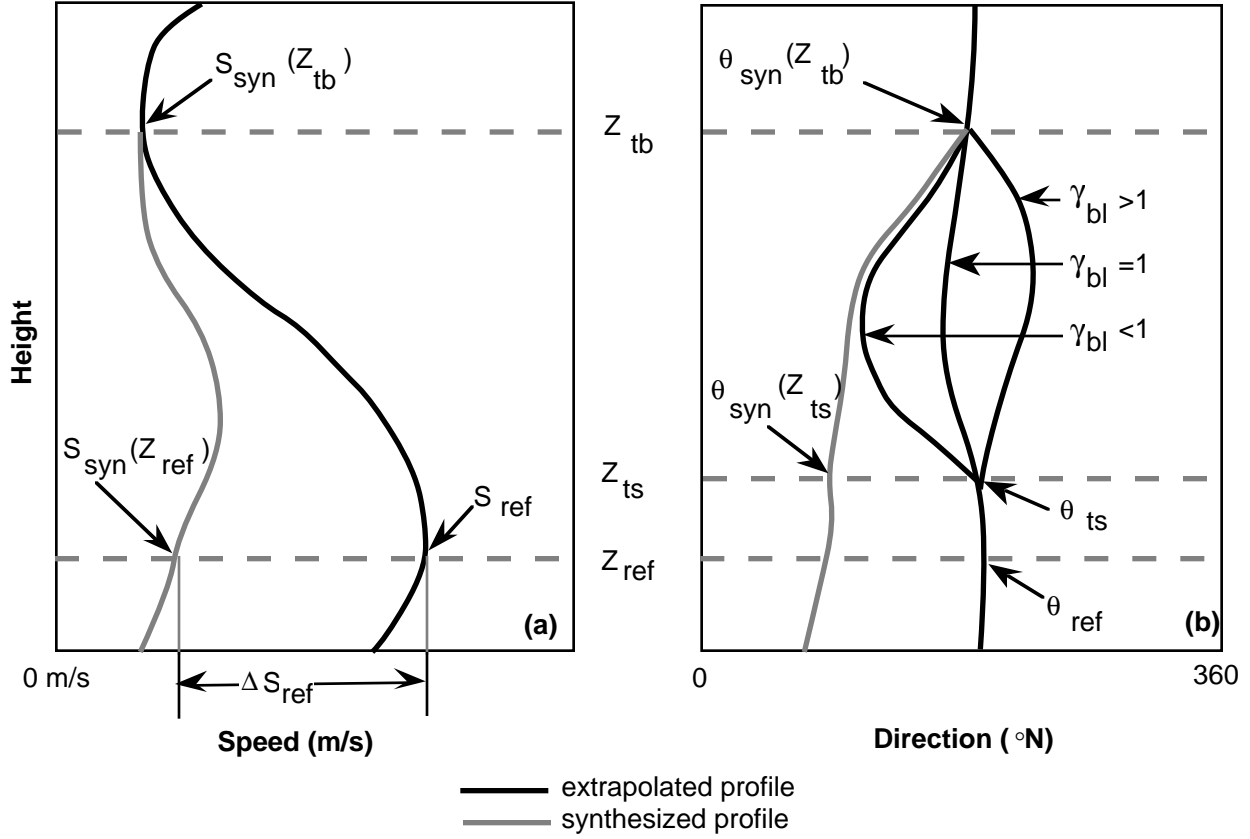
$$C_s(Z) = \frac{-\Delta S_{\text{ref}}}{Z_{\text{tb}} - Z_{\text{ref}}} (Z - Z_{\text{ref}}) + \Delta S_{\text{ref}} \text{ for } Z_{\text{ref}} \leq Z < Z_{\text{tb}} , \quad (\text{II.C.6a})$$

$$C_s(Z) = 0 \quad \text{for } Z \geq Z_{\text{tb}} , \quad (\text{II.C.6b})$$

where  $Z_{\text{tb}}$  and  $Z_{\text{ref}}$  are the heights of the boundary layer and the reference level, and  $Z$  is the height above terrain of the point under consideration.  $\Delta S_{\text{ref}}$  is the discrepancy between the reference level wind speed and the speed from the synthesized profile at the same height, i.e.,  $\Delta S_{\text{ref}} = S_{\text{ref}} - S_{\text{syn}}$  at  $Z_{\text{ref}}$ . The extrapolated wind speed for a grid point in the column are given by:

$$S(Z_k) = S_{\text{syn}}(Z_k) + C_s(Z_k) \quad \text{for} \quad Z_k \geq Z_{\text{ref}} \quad (\text{II.C.7})$$

where  $Z_k$  is the height of the grid point of interest. From examination of Eqs. (II.C.6) and (II.C.7), it can be seen that when  $Z = Z_{\text{ref}}$ , then  $C_s(Z_{\text{ref}}) = \Delta S_{\text{ref}}$  and  $S(Z_{\text{ref}}) = S_{\text{ref}}$ . When  $Z \geq Z_{\text{tb}}$ , then  $C_s(Z) = 0$  and  $S(Z) = S_{\text{syn}}(Z)$ . The correction term varies linearly between  $Z_{\text{ref}}$  and  $(Z_{\text{tb}})$ , and goes to zero at the top of the boundary layer above which the extrapolated profile merges with the synthesized profile. A sample extrapolation for a column of points is shown in Fig. II.C.6.



**Fig. II.C.6. Examples of (a) speed and (b) direction extrapolation (as a function of  $\gamma_{bl}$ ) using the profile method.**

The directional extrapolation procedure is similar to that for the speed. Upper air observations have no effect on the surface layer. Therefore, since there is no directional shear in the surface layer, the reference level direction,  $\theta_{ref}$ , is used for all grid points below  $Z_{ts}$  (it is always assumed that the reference height is within the surface layer). The effect of upper air soundings begins at the top of the surface layer and increases with height in the boundary layer until they completely dominate in the geostrophic regime. The directional shear in the intermediate boundary layer is controlled by the same parameters as in parameterized extrapolation ( $\gamma_{bl}$  and  $\theta_r$ ). The correction term is related to the discrepancy between the synthesized direction, based on upper air data, at the top of the surface layer and the reference level direction used throughout the surface layer. The correction term for direction is given by:

$$C_{\theta}(Z) = \frac{\Delta'\theta}{(\Delta Z_{bl})^{\gamma_{bl}}} (Z - Z_{ts})^{\gamma_{bl}} - \Delta'\theta_{ts} \text{ for } Z_{ts} \leq Z < Z_{tb} , \quad (\text{II.C.8a})$$

$$C_{\theta}(Z) = 0 \quad \text{for} \quad Z \geq Z_{tb} , \quad (\text{II.C.8b})$$

where  $\Delta Z_{bl}$ ,  $Z_{tb}$ ,  $Z_{ts}$ , and  $\gamma_{bl}$  have the same definitions as in the parameterized method,  $Z$  is the height of the point under consideration, and  $\Delta'\theta_{ts} = \theta_{syn}(Z_{ts}) - \theta_{ts}$  is the direction discrepancy at the top of the surface layer between the surface observations and the profile synthesized from the upper air data. The sense of how this discrepancy is resolved (via backing or veering) is controlled by

$\theta_r$  in a manner analogous to Eq. (II.C.5), except that  $\Delta\theta = \theta_{\text{syn}}(Z_{\text{ts}}) - \theta_{\text{ts}}$ . It is important to note that  $\theta_r$  in the profile method is used for comparing angles at the top of the surface layer, while  $\theta_r$  in the parameterized method is used to compare angles at two different vertical levels (the extremes of the boundary layer). The extrapolated wind for a grid point in the column is given by

$$\theta(Z_k) = \theta_{\text{syn}}(Z_k) + C_\theta(Z_k) \quad \text{for} \quad Z_k \geq Z_{\text{ts}} , \quad (\text{II.C.9})$$

where  $Z_k$  is the height of the grid point of interest. From examination of Eqs. (II.C.8) and (II.C.9), it can be seen that  $\theta(Z_{\text{ts}}) = \theta_{\text{ts}}$  and  $\theta(Z_{\text{tb}}) = \theta_{\text{syn}}(Z_{\text{tb}})$ . The correction term varies between  $Z_{\text{ts}}$  and  $Z_{\text{tb}}$  as a function of  $\gamma_{\text{bl}}$ . If  $\gamma_{\text{bl}} = 1$ , then the influence of the surface observations goes linearly to zero at the top of the boundary layer. If  $\gamma_{\text{bl}} \gg 1$ , the influence of the surface observations is maintained throughout most of the boundary layer before vanishing at the top. If  $\gamma_{\text{bl}} \ll 1$ , then the synthesized profile dominates the extrapolated winds through most of the boundary layer. The effects of selected choices for  $\gamma_{\text{bl}}$  are shown in Fig. II.C.6.b. When the process described above is completed for each column of points in the extrapolation grid volume, these winds can be examined by the assessor. As in the parameterized case, all the user-defined parameters in the process are available for adjustment if the extrapolated wind field fails to give an acceptable description of the atmosphere.

### II.C.2.c. Terrain Incorporation

In both methods the final products are grids of  $u$ - and  $v$ -component winds. The extrapolated wind arrays are transformed to the CG-MATHEW grid volume by the incorporation of terrain which is controlled by the TERRAIN\_ADJUSTMENT variable. If set to 'N' (the default for hemispheric MEDIC), the winds are zeroed up to the terrain level, defined as the highest of the (usually four) cell-centered terrain levels adjacent to a grid point column, as generated by TOPOG. If TERRAIN\_ADJUSTMENT is set to 'Y' (the default for regional MEDIC), each column of grid points is shifted upwards by the number of grid points along the column that are at, or below, the block terrain surface in the CG-MATHEW grid volume. Slip velocities are then assigned along vertical faces of the terrain surface using the  $u$ - and  $v$ -components of the slip velocity at the bottom of the extrapolation grid volume. Components that are tangential to a vertical terrain surface are assigned to the related grid point unless, at the same point, the component is normal to another vertical terrain surface. Once terrain incorporation has been performed, the winds are ready to be passed to CG-MATHEW.

### II.C.2.d. Gridded wind fields

Upper air data available in the form of AFGWC gridded metadata files may be used by MEDIC. The AFGWC files contain analysis or forecast data and cover either the northern or southern hemisphere to within 20 degrees of the equator. The upper air metadata data is in the form of  $u$ - and  $v$ - component gridded winds on constant pressure surfaces of dimension. The AFGWC grids are dimensioned 47 by 51 by 15. MEDIC outputs tables of the valid wind data for each AFGWC file



input. It is intended that other gridded data sources will be supported in the same way in MEDIC as these data become available.

#### **I.C.2e. Hemispheric MEDIC (HMEDIC)**

The hemispheric version of MEDIC is built on a 47 by 51 by 15 grid to match the AFGWC grids using a polar stereographic projection. Winds input from AFGWC upper air data (see Part II.C.2.d) are linearly interpolated to the hemispheric computational grid and the appropriate map projection correction is applied. At least two pressure levels of  $u$ - and  $v$ - data must be present for continued processing of the AFGWC winds from a given file. Otherwise a warning is printed and the code skips to the next data set. Terrain may be incorporated by zeroing winds up to, but not including, the terrain surface (see Part II.C.2.c).

### II.C.3. PLCNT

PLCNT is a code for processing the concentrations produced by ADPIC. The ADPIC output is passed to PLCNT in CONC\*.BIF (see Section III.C.10), a file that contains one concentration grid set for each sample bin in ADPIC. A concentration grid set includes a main grid, matching the ADPIC advection grid, along with up to five nested grids each with a dimension equal to one half the main grid and a grid step also equal to half the main grid step. A CONC\*.BIF file is created at each ADPIC output time, where the output interval is specified by SAMPLING\_INTERVAL.

Concentrations can be sampled in three ways: instantaneous, time integrated, and total deposition. (Future plans include addition of a fourth way: integrated deposition.) Contours of instantaneous air can be considered “snapshots” of concentration (or dose) at a given point in time and at a specific level above the surface. Contours of integrated air depict the total amount of pollutant passing a given point during a specified time interval. If a dose conversion factor is specified, then the contours represent the total dose to an unshielded human from exposure to a plume or puff. Contours of deposition show distributions of material deposited since the beginning of the release to the time of output. In an ARAC context, results are generally expressed as micrograms per square meter ( $\mu\text{g}/\text{m}^2$ ), curies per square meter ( $\text{Ci}/\text{m}^2$ ), or a dose to the skin or through inhalation or ingestion.

To produce a desired plot, the user must appreciate the interplay between certain parameters in ADPIC and PLCNT. The critical parameters in ADPIC are SOURCES\_TO\_SAMPLING\_BIN (the sources to be associated with a sampling bin, by number), SAMPLING\_TYPE (the type of concentration sampling), and SAMPLING\_HGT (the sampling height for integrated and instantaneous air sampling). These arrays are all dimensioned maxsbins (currently 9) with corresponding elements of each array referring to one of the concentration grid sets (i.e. a main grid and up to four nested grids) that define a sampling bin in ADPIC. Thus, the first element of SOURCES\_TO\_SAMPLING\_BIN indicates which of the MAXSOURC sources is to be considered in the first ADPIC bin; the first element of SAMPLING\_TYPE indicates whether the first ADPIC sampling bin is to be integrated, instantaneous, or deposition; the first element of SAMPLING\_HGT indicates the height above terrain at which the first ADPIC sampling bin is to be placed. The second elements of each array describe the second sampling bin, and so on.

PLCNT can be thought of as having a dose or units conversion function and a contour plotting function. The three parameters that control the dose conversion are SAMPLING\_TO\_DOSE\_BIN, DCON, and DOSE\_TO\_FINAL\_BIN, which are contained in the CONC\_ANALYSIS.NML file. The first two parameters allow the user to combine bins to produce combined nuclide dose plots (e.g., combined dose from multiple nuclides or multiple hybrid sources), and to re-use bins to produce different contour plots from the same ADPIC sampling bin (e.g., thyroid dose and effective WB dose derived from integrated air concentration). SAMPLING\_TO\_DOSE\_BIN specifies how the sampling bins are to be assigned to the dose bins; DCON specifies the dose conversion factors and/or scaling

factors to be applied to the dose bins (also see description for DCON in Section III.A.8); DOSE\_TO\_FINAL\_BIN specifies how the dose bins are to be assigned to the final bins. Examples and their translations are shown in Figure II.C.7.

A	{	SPECIES	= 'I-131'	'Kr-88'	'Pu-239'
		SOURCES_TO_SAMPLING_BIN(1,1)	=1		
		SOURCES_TO_SAMPLING_BIN(1,2)	=2		
		SOURCES_TO_SAMPLING_BIN(1,3)	=3		
		SAMPLING_TYPE	=1	1	1
B	{	SAMPLING_HGT	=1.5	1.5	1.5
		SAMPLING_TO_DOSE_BIN	=1	1	2
		DCON	=3.60E2	7.63	3.65E-1
C	{			3	3
				1.13E5	3.96E5
		DOSE_TO_FINAL_BIN(1)	=1		
		DOSE_TO_FINAL_BIN(2)	=2	3	4
	{	DOSE_TO_FINAL_BIN(3)	=5		

Translation of A:

Sampling bin #1 will contain integrated (at 1.5m height) air concentration of I-131 source.  
Sampling bin #2 will contain integrated (at 1.5m height) air concentration of Kr-88 source.  
Sampling bin #3 will contain integrated (at 1.5m height) air concentration of Pu-239 source.

Translation of B:

Dose bin #1 will contain sampling bin #1 converted to I-131 Thyroid dose via inhalation.  
Dose bin #2 will contain sampling bin #1 converted to I-131 Effective WB dose via inhalation.  
Dose bin #3 will contain sampling bin #2 converted to Kr-88 Effective WB dose via immersion.  
Dose bin #4 will contain sampling bin #3 converted to Pu-239 Effective WB dose via inhalation.  
Dose bin #5 will contain sampling bin #3 converted to Pu-239 lung dose via inhalation.

Translation of C:

Final bin #1 will contain dose bin #1 (data for thyroid dose plot from I-131).  
Final bin #2 will contain dose bins #2, #3, #4 (data for Effective WB dose plot for three nuclides).  
Final bin #3 will contain dose bin #5 (data for lung dose plot from Pu-239).

**Fig. II.C.7. Example of interplay between ADPIC and PLCNT parameters to produce desired types of plots.**

The times of contour plot generation are determined by defining one or more plot series for each final bin of interest. A plot series has a beginning and an ending point in time along with a time step that specifies how often a plot is to be generated. The first plot is produced at the beginning of the plot series (which must match the time of a CONC\*.BIF file) and subsequent plots in the series are produced at equal intervals after that until time steps beyond the end of the plot series. In the case of interval integrated air plots, an integration step may be specified that may be less than the plot step.

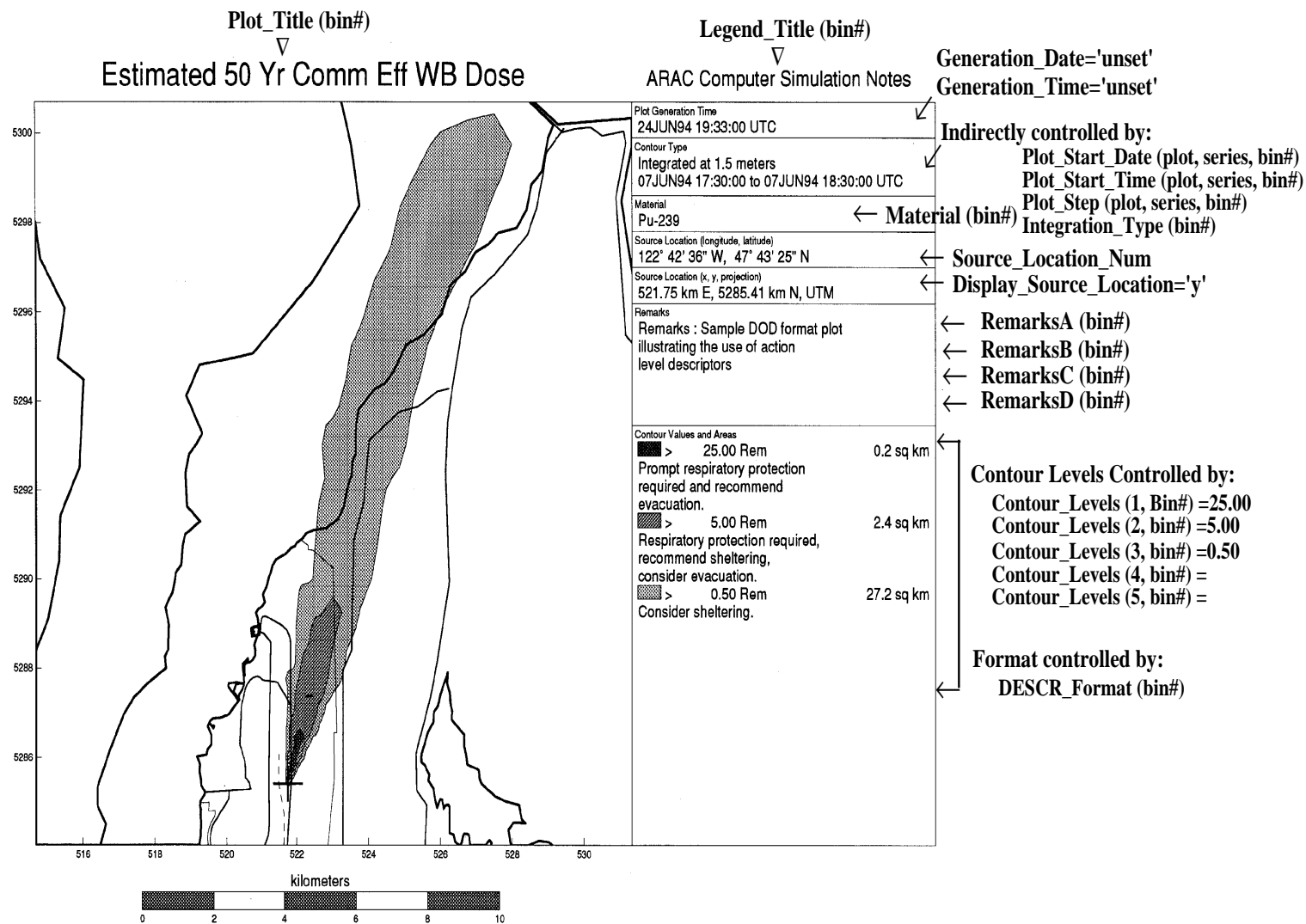
In such cases, the integrated plot will include concentrations beginning one integration step before the end of the current plot step, and thus, the integration period will end at the time of the next plot. Plot series are started with the variables PLOT\_START\_DATE/TIME and explicitly ended with PLOT\_END\_DATE/TIME. If a PLOT\_END\_DATE/TIME is not specified then the current plot series is ended when another plot series for the same final bin begins or by the end of the available concentration data in the series of CONC files. The plot step is specified by PLOT\_STEP and the integration step by INTEGRATION\_STEP. All these variables are described in more detail in Section III.B.9.

PLCNT allows the user to select up to maxcontlevel (currently 5) contour levels using the CONTOUR\_LEVELS variable (see Section III.B.9). PLCNT will also generate contour levels in the absence of user-specified values. The code will select the largest value of the form 1.0En or 3.0En that is lower than the highest grid point concentration value for a given final bin (if HALF\_POWER\_OF\_10 is set to 'N' only values of the form 1.0En will be selected). This becomes the value of the innermost contour. For example, if the maximum grid point concentration is 6.9E8, then the innermost contour will be 3.0E8. If the maximum grid point concentration is 2.4E8, then the innermost contour will be 1.0E8. Up to four more contours, each decreasing by an order of magnitude, are then computed to fill out the maximum desired number of contours (specified by MAX\_CONTOURS). The above example strictly applies only if no restrictions are made on the minimum area to be covered by the maximum contour. (See the description of MIN\_CONTOUR\_AREA in Section III.B.9 for more information). If the user selected contour values but the concentrations are too small to produce a contour with area greater than MIN\_CONTOUR\_AREA, then PLCNT will optionally compute contours based on the maximum concentration (see RECOMPUTE\_USER\_CONTOURS and NUM\_RECOMPUTED\_CONTOURS in Section III.B.9).

ADPIC produces concentrations at grid points. PLCNT smooths the doses maintained in the final bins along the edges of the nested grids, thereby incorporating some influence of the values in the coarser grid surrounding the nest. This avoids almost all discontinuities in the contour lines when they cross nest boundaries. This smoothing is not required for the X-Windows versions concentration plotting codes and therefore the site workstation products are written from the final bins before these bins are smoothed.

Since ARAC provides a service to more than one customer, and each customer has slightly different needs, limited flexibility of plot layout has been allowed for, primarily focused on the lower half of the plot legend. In addition, the exact wording contained in almost all of the legend is also under user control. The layout and wording of the contour plot is controlled by a multitude of parameters. Figs. II.C.8, II.C.9, and II.C.10 show examples of three of the more common layouts and the values of the PLCNT parameters which control the resulting appearance.

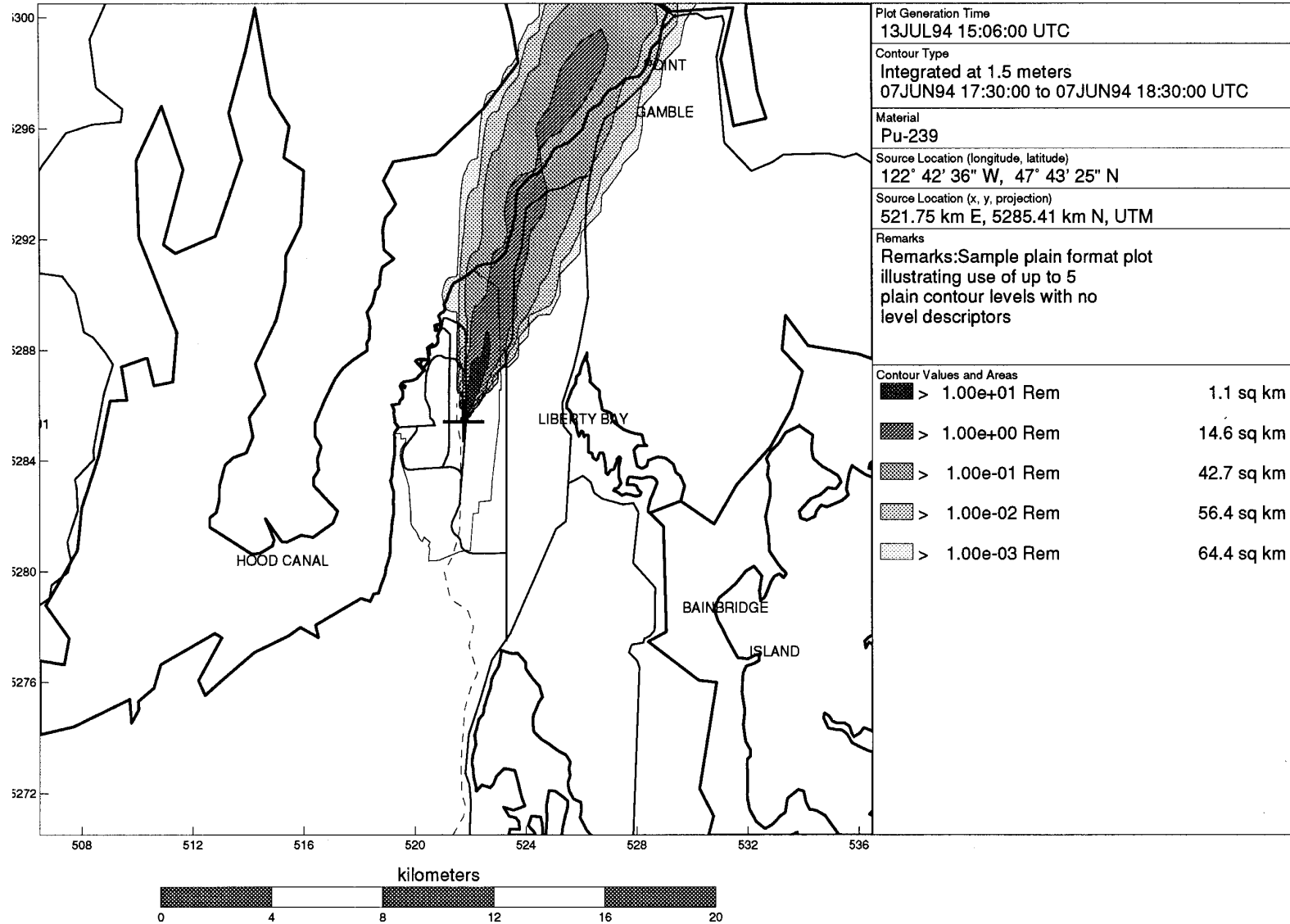
**Fig. II.C.8. Example of plot layout for DOD plots and PLCNT parameters which control the layout and wording.**



# Estimated 50 Yr Comm Eff WB Dose

## ARAC Computer Simulation Notes

Fig. II.C.9. Example of plot layout for plain plots and PLCONT parameters which control the layout and wording.



# Integrated Air Concentration

## ARAC Computer Simulation Notes

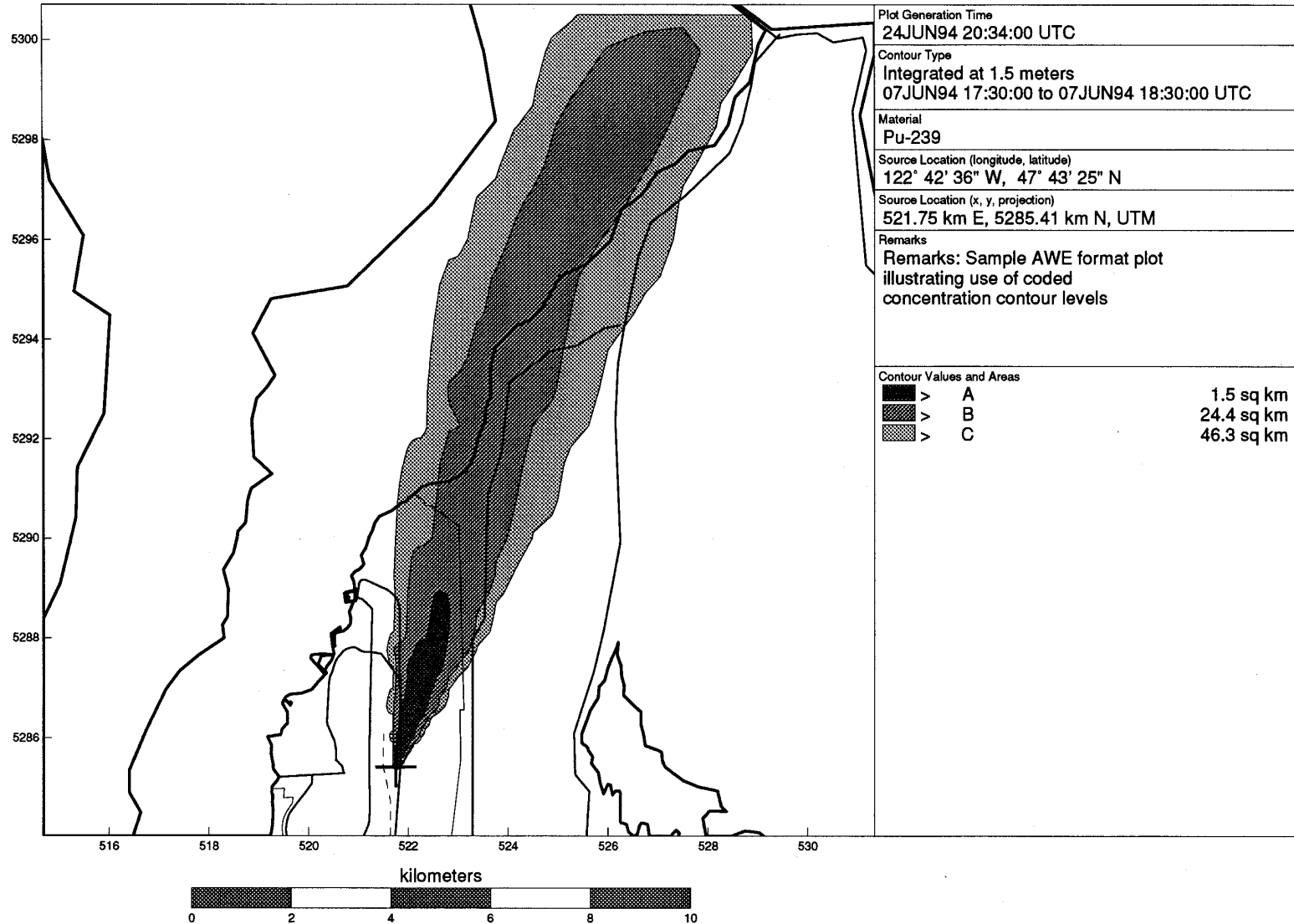


Fig. II.C.10. Example of plot layout for AWE plots and PLCNT parameters which control the layout and wording.

The ability to specify the area to blow up (or “window”) for plotting is provided by the XMIN\_WINDOW, XMAX\_WINDOW, YMIN\_WINDOW, and YMAX\_WINDOW parameters. These four parameters specify the coordinates of the area to be blown up. Be aware that the areas specified in the legend of a blown-up plot refer to the areas covered by the contours in the original-size plot.

The ability to automatically generate a mosaic of subplots covering a specified area at a specified scale is provided by the following parameters: MAP\_SCALE\_FLAG, MAPSCALE, SOURCE\_OCTANT, BLOWUP\_SOURCE, PLOT\_SIZE\_CM. Up to 225 subplots (15 by 15) can be generated for a specified scale and output device. The placement of the ADPIC source location in the subplot containing it is controlled by SOURCE\_OCTANT. The area to be split into subplots can be narrowed down by use of the blowup parameters mentioned in the previous paragraph.

The centerline capability of PLCNT starts at a user-specified ADPIC source location and searches along arcs of increasing radius for the maximum along each radial arc. The resultant series of maxima, along with their locations are available in tabular format in the CENLINE\*.DAT file, and the locations are plotted on each Site System plot. All centerline parameters are contained in the CONC\_ANALYSIS\_CONTROL namelist: CENTERLINE\_SOURCE is the only required parameter; optional parameters include RADIAL\_STEP\_FACTOR, FULL\_SEARCH\_FACTOR, ANGLE\_WIDTH\_FACTOR, SECTOR\_ANGLE\_MIN, and SECTOR\_ANGLE\_MAX.

#### **II.C.4. TIMEHIS**

The Time History (TIMEHIS) code is an offspring of the PLCNT code. Concentration values are binlinearly interpolated for specific point locations. The times of the interpolations are controlled in the same way as in PLCNT and are tabulated in the TIMEHIS.LOG file as well as be plotted on a semi-log plot (log concentration vs. time). The number of data points per plot is controlled by the TIMESTEPS\_PER\_PLOT parameter. Interpolated points with values of zero are considered to be  $1.0\text{e-}30$  by the plotting routine so those values are plotted off scale to the bottom of the logarithmic vertical scale of the time history plot.

If actual measurement values are available, these values can also be input via the MEASUREMENTS section of the SAMPLER.DATA file, and appear on the plots along with the model-calculated values for the specified locations and times. Some additional statistics, based on the most straightforward ones used in the ATMES study, are also performed to compare the computed and measured concentrations. In addition to the TIMEHIS.LOG, which contains listings of the same time history data that is plotted, TIMEHIS produces two other ASCII files: SAMPOUT.DAT and STATS.DAT. SAMPOUT contains the interpolated value at each point along with the minimum and maximum point concentrations found within angles of various sizes (hit angles) of the main interpolation point. That is, concentrations are interpolated at a grid of points (currently 11 by 11)



ranging from  $-dtheta$  to  $+dtheta$  about angle  $theta$  and  $-dtheta*radius$  to  $+dtheta*radius$  centered on  $(radius,theta)$  where  $radius$  and  $theta$  are the polar coordinates of the main interpolation point with respect to a user selected source location and  $dtheta$  is specified hit angle (see HIT\_ANGLES and STATISTICS\_SOURCE in Section III.B.9). These minima and maxima are compared against measurements and if the measurement lies between the two values it is considered an exact match at that angle. The smallest hit angle is considered to be zero so the main point interpolated value is found listed in the first entry in the hit angle table ( the minimum and maximum will always be the same for this entry). The STATS.DAT file contains the hit angle statistics along with some other simple statistics based on those used in ATMES.

#### II.C.5. RSUS (not implemented yet)

RSUS is a supplemental code which uses a given source term and geometry (via RSUSGEO) and user input (via RSUSIN) to produce a complete set of input files for a CG-MATHEW/ADPIC calculation that will simulate the resuspension, transport, and diffusion of particles deposited by a previous CG-MATHEW/ADPIC calculation (Porch *et al.*, 1983; Mitchell *et al.*, 1985).

When the option RESUSPEND = 'Y' is inserted in CONC\_ANALYSIS.NML file, PLCNT will sum the deposition concentrations of the cells that are greater than 10% of the maximum deposition concentration. The geometry of the area specified by these cells is then represented by a rectangle, with the distribution of particles within this rectangle assumed to be Gaussian. The source term geometry extends in the vertical from 0 m to 2 m with a standard deviation of 1 m. The source term concentration and geometry is contained in RSUSGEO, an output file created by PLCNT. User input (supplied by RSUSIN) needed to compute the vertical particulate flux includes the surface type (sandy, loam, clay or snow), drag coefficient, and the surface wind speed. The deposited toxic particles are assumed to be attached to the ground material to some degree, so that the resuspension of the toxic particle is based upon the resuspension of the ground material. The check for the setting of various “fine tuning knobs” is the resuspension factor (  $K$  ) defined by Fish (1964):

$$K(m^{-1}) = \frac{\text{airborne concentration (units/m}^3\text{)}}{\text{contamination level (units/m}^2\text{)}} .$$

Page 64 of Fish (1964) gives a resuspension factor range of  $10^{-4}$  to  $10^{-6}$  for PU239 obtained from experimental results of nuclear weapon trials at Maralinga. The resuspension factor is calculated by RSUS and is contained in the RSUS file. Its order of magnitude should be checked before making the resuspension CG-MATHEW/ADPIC calculation.

The length of the resuspension run is one hour, which is arbitrary. It should take weeks before all of the deposited material is resuspended, so there is no problem with running out of deposited material. But there is the problem of ADPIC's inability to simultaneously simulate the original cloud

and resuspended material, i.e., the material is continually being deposited and resuspended, but we can only follow one source at a time.

RSUS creates several files (STNLOC.MET, OBSERV.MET, TOPOG.NML, MEDIC.NML, ADPIC.NML, PLCNT.NML) for use in the resuspension CG-MATHEW/ADPIC calculation. You may change the input files to suit your scenario, but you should be aware of the following caveats.

The TOPOG.NML file is set up such that the source is located near (about four grid cells) the edge or corner of the ADPIC grid so that as much of the plume will be displayed on the plot as possible. The grid size may be changed if desired, but remember that any changes will affect the correction factor for the average air concentration (thereby affecting the resuspension factor) if you are running with the non-nested version of the codes.

The OBSERV.MET file is set up for the surface wind speed and direction specified in RSUSIN, has no speed or direction shear in the vertical, and is for neutral stability. You can also edit this file, but be forewarned that the experimental data upon which resuspension source term is based is for neutral stability, and that the magnitude of the resuspension source term is dependent upon the surface wind speed specified in RSUSIN. Surface and upper air metdata from other regional stations may be added to the OBSERV.MET, but it should be consistent with the surface wind speed and direction specified for the resuspension source location.

The cloud parameters and source rate in ADPIC should not be changed. The data available for plotting should also include an integrated plot for air concentration in  $g^* \text{ sec/m}^3$  at a height of 2 m to use as a double check for the value calculated by RSUS. Plots for integrated lung dose and total deposition are also specified since these are the usual plots requested. The SAMP\_CELL\_DELZ\_FRACT parameter is the ratio of the vertical sampling grid size (which should be 2 m, since this is the height used for calculating the resuspension factor) to the vertical advection grid size (e.g., SAMP\_CELL\_DELZ\_FRACT = 0.04 for a 50 m vertical advection grid size).

The PLCNT is set up for the above-mentioned three plots. The integrated lung dose DCON was calculated as follows:

$$DCON = (8.02E - 2 \text{ Ci/g}) \times (4.0E5 \text{ Rem}^* \text{ m}^3/\text{Ci/sec}) = 3.21E4, \text{ UNITS} = \text{Rem}.$$

The total deposition DCON was calculated as follows:

$$DCON = (8.02E - 2 \text{ Ci/g}) \times (1.0E6 \mu \text{ Ci/Ci}) = 8.02E4, \text{ UNITS} = \mu \text{Ci/m}^2.$$

The PLOT\_STEPS were arbitrarily set for 15-minute interval plots.

## II.D. CG-MATHEW/ADPIC Validation and Error Analysis

Because of the expense in computer time, the complexity of the three-dimensional transport and diffusion models, and the ensuing large number of variable parameters involved, no comprehensive error analysis of CG-MATHEW/ADPIC has yet been undertaken. However, a number of tracer validation studies have been conducted and others are still in progress.

The codes were validated against closed, Gaussian solutions to the diffusion-advection equation (Lange, 1973), including simple wind shear. The standard deviations were found to be accurate to within 5%. Over the last twenty years the codes were also evaluated against numerous field tracer release experiments conducted under a variety of meteorological and topographical conditions (Foster and Dickerson, 1990). Neutrally buoyant tracers released both as surface and elevated point sources have been studied. A summary of the results is shown in Fig. II.D.1.

The results show that the CG-MATHEW/ADPIC models generally estimate the tracer air concentrations to within a factor of two of the measured values 20% to 50% of the time, and within a factor of five of the measurements 35% to 85% of the time depending on the complexity of the meteorology and terrain, and the release height of the tracer. These results are achieved without “tuning” the model to any given site, type of source, or sampling method.

While a detailed sensitivity analysis of CG-MATHEW/ADPIC has not been undertaken, a consensus has nevertheless emerged as to the degree of model sensitivity toward certain important variables. A side from numerical parameters, which are discussed in the technical descriptions of the codes, the main parameters affecting the quality of CG-MATHEW/ADPIC results, in order of decreasing importance, are:

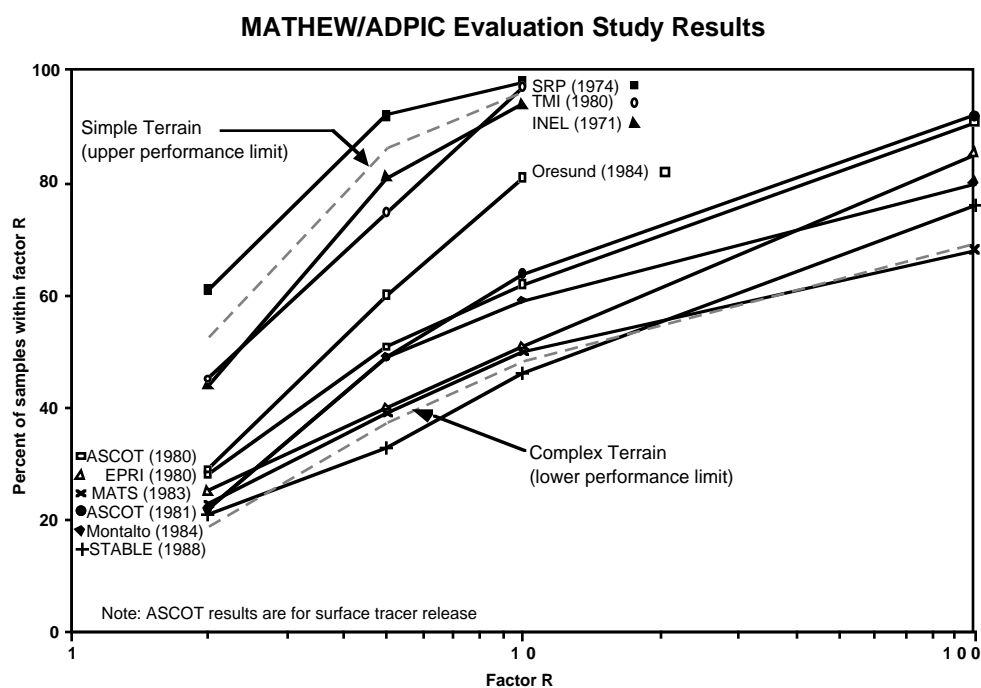
Wind direction. A difference of 10 degrees in wind direction can have a substantial effect on the agreement of model concentrations with individual sampler measurements. This is especially true on small horizontal scales (e.g., 10 km). Errors in model calculations are commonly caused by the lack of a sufficiently large set of meteorological (wind) measurements for extrapolation to CG-MATHEW's grid. Also, we must contend with the fact that the NOAA and FAA only report wind directions to the nearest ten degrees.

Topography. There is close interaction between topography and the wind field, especially near the surface. Because of the limitations on the number of computational cells available, the resolution of important topographical features, such as riverbeds, ridges, and canyons, is frequently poor. This is most severe in complex terrain situations and under stable atmospheric conditions.

Diffusion parameters. Most turbulent diffusion theory is semi-empirical, even for homogeneous turbulence. To describe diffusion parameters in space and time for all but the most trivial atmospheric conditions is very difficult. Factors of two to five in concentration differences computed by CG-MATHEW/ADPIC can appear because of uncertainties in diffusion parameters.

Wind speed. While much less likely to produce errors than the wind direction, the wind speed can, nevertheless, produce errors of factors of two quite easily. Model errors due to wind speed are harder to detect because they are difficult to separate from errors caused by the diffusion parameters.

Source strength. In controlled experiments the source strength is generally well known; however, in accident situations this is not always true. During an emergency response to an accident, the uncertainty of the source term and the properties of the source material can introduce significant errors into the model results.



STUDY	YEAR	LOCATION	TOPOGRAPHY	ATMOSPHERIC STABILITY	RELEASE HEIGHT
INEL	1971	Idaho Falls, ID	Flat-Rolling	Unstable	Surface
SRP	1974	Aiken, SC	Flat-Rolling	All	Surface, Elevated
ASCOT	1980	The Geysers, CA	Complex	Stable	Surface, Elevated
ASCOT	1981	The Geysers, CA	Complex	Stable	Surface, Elevated
TMI	1980	Harrisburg, PA	Flat-Rolling	All	Elevated
EPRI	1980-1	Kincaid, IL	Flat	All	Elev + Plumerise
Oresund	1984	Oresund Strait, Denmark/Sweden	Flat-Rolling	Neutral (spatially transitional)	Elevated
MATS	1983	Aiken, SC	Flat-Rolling	Neutral, Unstable	Elevated
Montalto	1984	Italy	Flat-Rolling	Unstable	Surface, Elevated
STABLE	1988	Aiken, SC	Flat-Rolling	Stable	Elevated

**Fig. II.D.1. Percentage of cases in which CG-MATHEW/ADPIC results are within a factor N of field data.**



### **III. User's Guide**

#### **III.A. ARAC Models File System and Flowcharts**

Two classes of information need to be passed along the ARAC model sequence. First, the major results of each model must be passed to the next model, mainly as various arrays of data (e.g., model elevations, winds, concentrations). Second, parameters used by one model that are needed by subsequent models for purposes of processing and labeling. There is also a need to maintain a record of the information used in a given model execution to support precise analysis and debugging, i.e., a traceback capability is necessary. The ARAC models file system provides for these three requirements. This section will discuss the file system (both naming conventions and content) which allows the ARAC models to handle these three functions.

The first component of the file system is the (invisible) TraceBack Block (TBB) that appears at the beginning of every file used to pass data between models. It consists of directory and file names that indicate the data files used to generate the associated model results contained in the remainder of the file. For this traceback function to work, each file referenced by the TBB must be uniquely named. These unique names include qualifiers and sequence numbers as described later. The input files indicated may also be read by subsequent models as the source of any parameters required for that model's execution. Thus the only data transferred via binary intermodel files (BIFs) are actual model results. The exception is grid/topography data produced by TOPOG. This information will be read by all the models to define the grid geometry and so occupies a unique place with respect to the rest of the models. The grid definition file (TOPOG\*.GRID) contains the grid parameters as well as the model topography cell heights (here we use a convention convenient in discussing these uniquely qualified files: TOPOG\*.GRID means all qualified files beginning with TOPOG and ending with the suffix .GRID. Similarly, \*.NML means all files which end with the suffix .NML; in addition to files that are required to be uniquely names to support traceback, some other types of files also conform to this convention for consistency, e.g., \*.CGMB files). Due to its central place in the model scheme and its small size the TOPOG\*.GRID file is maintained in ASCII and can be viewed like any other such file.

Each model is controlled by a namelist file (\*.NML) which contains the parameters controlling the model run. These files are read by the appropriate model from generically named files (e.g., TOPOG.NML, MEDIC.NML) after which the information is rewritten to a uniquely named completed namelist file (\*.CNML) that will contain the values of all variables in the namelist (including data-loaded defaults) in a namelist-readable format. Subsequent models read these CNML files as needed. Almost all output files contain one data set per file. That exception is the \*.LOG files which are intended to record a log of a particular model run and not remain as part of the files documenting the run. \*.LOG files have version numbers to distinguish them, but one the ARAC VMS systems multiple files of the same name and a different version number are purged on a frequent basis. Thus, uniquely named files survive a system initiated purge while \*.LOG files do not. A single ADPIC run generally processes a number of MEDVEL/MATVEL files. To document this, a file listing

the MEDVEL/ MATVEL files used during the creation of a specific concentration output file (CONC\*.BIF) in generated (CONC\*.VLST) to pair with it. Similarly, PLCNT and TIMEHIS write files (\*.CLST) which documents the CONC files used in the program execution. These lists of data files provide a starting point in the event a complete traceback of an ADPIC, PLCNT or TIMEHIS run should be necessary.

There are three main components that comprise the technical details of this system: 1) file naming conventions, 2) the TBB structure, and 3) a TBB definition capability that allows substantial flexibility in permitting essentially unbounded expansion of the models in terms of the number of files read and written. The descriptions of the last two of these components are implementation details intended largely for programming staff and are described in Appendix C. The rest of this section discusses the file naming conventions.

As mentioned above, the traceback system relies on unique file names for all files maintained as part of it. The need for unique file names and some descriptive information on the file's contents have been combined to create human-readable file names. To assist understanding and file manipulations, the filenames follow one of three standard formats distinguished by the qualifier field. The various fields in a name of a given format type will tend to line up when listed with a directory search command. This feature helps both man and machine to see 'at a glance' what each file contains. The general file name structure has the following stencil:

prefix	qualifier	sequence	extension
7	24	4	4/5

The *prefix* in combination with the extension indicates specific file type. If the file extension is sufficient to identify the file, the name of the model that created the file is used as the prefix. Graphics output files are of this type, for the suffix .CGMB uniquely identifies the file type and the originating program's name is needed for graphics processing.

The *qualifier* identifies the data set as uniquely as possible given the space constraints of the filename. There are three distinct qualifier formats. The first is used with the TOPOG\*.GRID files, which contain grid information. The second format includes time information appropriate to the file and applies to most other files in the system. These qualifiers include a starting date and time along with duration or averaging time as appropriate. For clarity, three character alphabetic months are used in dates. The last format uses the first 12 characters of both the problem and run names to qualify the file name. This last type is useful for STNLOC.CMET, where the selection of station locations being output for reference is not dependent on time, but rather the location or region used in the problem. Thus, the problem and run names are sufficient to be unique qualifiers.

<u>GRID files:</u>	rrrrrkmzzzzzmxxxxx_yyyy
rrrrr	= the larger of east/north range in rounded km
zzzzz	= the vertical grid range in m
xxxxx	= the x coordinate of the grid center in rounded km
yyyyy	= the y coordinate of the grid center in rounded km

underscores = unassigned characters

date/time files: ddmmyy\_hhmm\_ss\_ddd\_hhmm

dd	=	the date w/ leading zeros
mm	=	the month as three alphabetic characters
yy	=	the year w/ leading zeros
ddd	=	duration in days w/ leading zeros
hh	=	the hour or number of hours of duration w/ leading zeros
mm	=	the minute or number of minutes of duration w/ leading zeros
ss	=	the seconds or number of seconds of duration w/ leading zeros

NOTE: For conc\*.bif and conc\*.vlst files, the ddd\_hhmm field indicates time from start of run while for most other files that field gives an averaging time or valid time interval.

time-independent files: ppppppppppppprrrrrrrrrrrr

pppppppppppp	=	the first 12 characters of the problem name
rrrrrrrrrrrr	=	the first 12 characters of the run name
underscores	=	unassigned characters

Having a *sequence* number which is always changing (increasing) forces unique file names with repeated model executions with the same or similar data. It is always one larger than the highest sequence number of any previously existing file. The sequence number is normally displayed as three digits with a leading underscore for clarity; however, if the highest sequence number of an existing file equals or exceeds 999 then the underscore prefix is usurped to allow sequence numbers up to 9999.

The *extension* indicates the class of files as follows:

.NML	Input namelist file
.CNML	Completed namelist file
.MET	Metdata input file (observations or locations)
.CMET	Completed metdata file (observations or locations)
.DATA	Sampler data input file
.CDAT	Completed sampler data file
.BIF	Binary Intermodel file
.CGMB	Computer Graphics Metafile binary format
.GRID	Model grid file
.VLST	Velocity filename list
.CLST	Concentration filename list
.TBB	PLCNT/TIMEHIS ASCII traceback block
.LOG	Model run log file

The following is a list of the files to be used by the main model system, grouped by the relevant model:

TOPOG.NML  
TOPOG\_rrrrrkmmzzzzmxxxxx\_yyyynnn.CNML



TOPOG\_\_rrrrkmzzzzmxxxxx\_yyyynnn.CGMB  
TOPOG\_\_rrrr kmzzzz mxxxxxyyyy\_nnn .GRID  
TOPOG.LOG

MEDIC.NML  
MEDIC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CNML  
STNLOC.MET  
STNLOC\_pppppppppppprrrrrrrrrrr.CMET  
OBSERV.MET  
OBSERV\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CMET  
MEDIC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CGMB  
MEDVEL\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .BIF  
MEDIC.LOG  
MEDIC.VLST

CG-MATHEW.NML  
CG-MATHEW\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CNML  
CG-MATHEW\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CGMB  
MATVEL\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .BIF  
CG-MATHEW.LOG  
CG-MATHEW.VLST

ADPIC.NML  
ADPIC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CNML  
ADPIC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .RST  
ADPIC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CGMB  
CONC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .BIF  
CONC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .VLST  
CONC\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .RST  
PARTPOS\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .BIF  
ADPIC.LOG  
ADPIC.CLST

CONC\_ANALYSIS.NML  
PLCNT\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CNML  
PLCNT\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CGMB  
PLCNT\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CLST  
PLCNT\_\_ddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .TBB  
PLCNT.LOG

TIMEHISddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CNML  
SAMPLER.DATA  
SAMPLERddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CDAT  
TIMEHISddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CGMB  
TIMEHISddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .CLST  
TIMEHISddmmmyy\_hhmm\_ss\_ddd\_hhmm\_nnn .TBB  
TIMEHIS.LOG

Figure III.A.1 illustrates the flow of the overall CG-MATHEW/ADPIC model system. The step-by-step progression through each of the five executable programs is shown by bolded arrows. (The TIMEHIS program is only used in conjunction with sampler measurement data.) Figures III.A.2 - III.A.4 show in greater detail the files associated with each of the five programs.

The ARAC models can be run on a variety of map projections, which can be selected to minimize the projection-induced scale distortion or to match a projection chosen by an external organization. In addition, to match the details of the geometry of a particular study area it is possible to reposition the origin of a projection as well as to rotate the projection. The models are executed on the current map projection as defined by the logical MAP\$F\_PROJECTION\_PARAMETERS. This logical points at an ASCII file that defines a projection.

Not all map projections are appropriate for modeling with the ARAC models. The applicable projections are conformal, which implies that that a single map scale applies at a given location and that relative direction is unaffected by the projection. The conformal projections currently implemented and available for use in the modeling system are the UTM, Ordnance Survey and GWC North and South specific projections and any normal Mercator, transverse Mercator, Lambert conformal conic, polar stereographic or general stereographic projection. If a model is run with a projection, the wind directions and speeds are corrected in MEDIC to account for the projection-induced distortion. The projection information is also used in the model graphics to indicate the grid orientation via a north arrow or a graticule. As a result, all the models need to know the current projection. The choice of whether to use a projection or not is made in TOPOG with the namelist variable MAP\_ADJUSTMENT. If a map projection is not used the winds are not adjusted in MEDIC and no grid orientation indication is made on the graphics.

The step of choosing a map projection takes place in production and Beta when a new problem is created using the Select\_Map\_Projection and Create\_Map\_Projection commands. The map projection file is placed in a problem-specific directory that is referenced with the logical MAP\$A\_PROJECTION\_PARAMETERS. Outside of the operational system, the user must ensure that the MAP\$F\_PROJECTION\_PARAMETERS and MAP\$A\_PROJECTION\_PARAMETERS logicals are set to point at the appropriate projection definition file. for additional details, please see the document "The ARAC Map Projection System".

The input files TOPOG.NML, MEDIC.NML, CG-MATHEW.NML, ADPIC.NML, and CONC\_ANALYSIS.NML contain model parameters in Fortran namelist format. These namelist input files are created and modified by the user. Each executable produces a log file (primarily a journal file used for debugging purposes) and a graphics file, as well as other files which are described in this section.

TOPOG uses model parameters from TOPOG.NML and terrain data to create the underlying model grid cell heights, which are written to the GRID file.

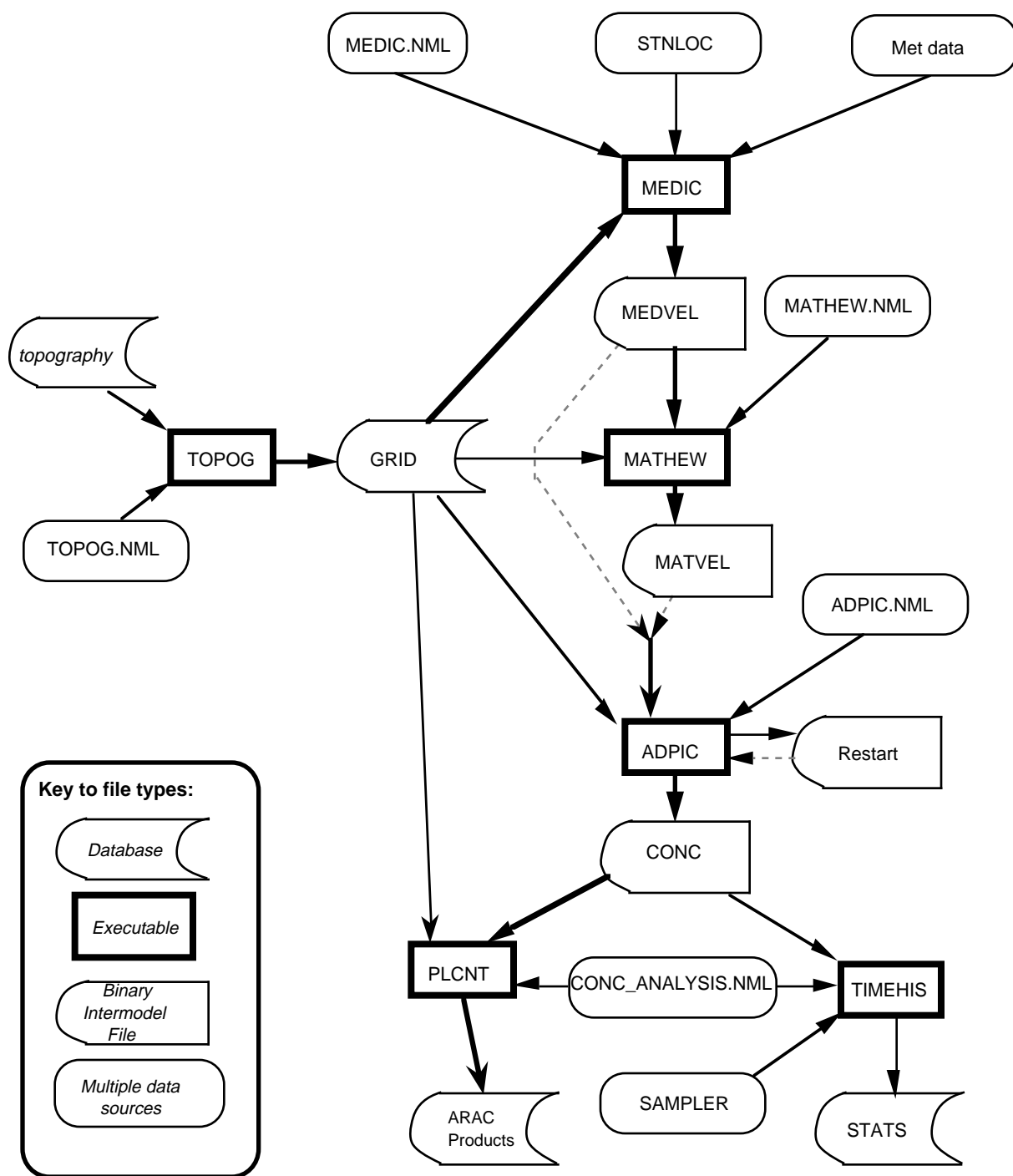
MEDIC uses model parameters from MEDIC.NML to select meteorological data from various sources and to specify parameters used in interpolating and extrapolating that data. MEDIC can

access met station location either from a Problem Station Library or from a STNLOC.MET namelist file, and meteorological data can be accessed either from Problem Metadata Files or from an OBSERV.MET file. The former methods for met station information and meteorological data are the primary methods used by ARAC operational responses, while the latter methods are used primarily for canned metadata cases. MEDIC produces MEDVEL files, one for each met dataset, containing the three-dimensional gridded wind fields.

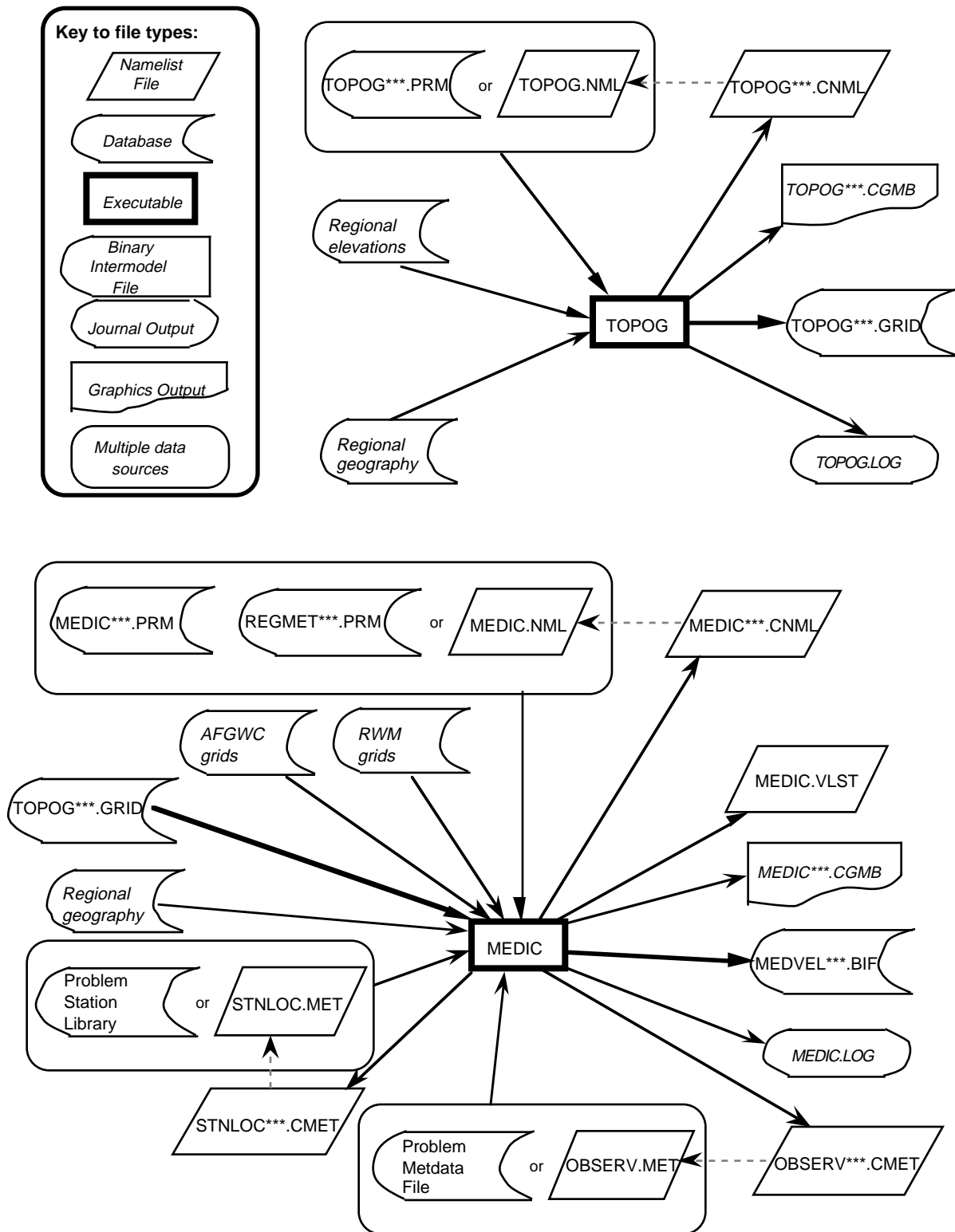
CG-MATHEW uses the model parameters from the CG-MATHEW.NML file to adjust the gridded wind fields from the MEDVEL files to make them mass-consistent. These mass-adjusted three-dimensional wind fields are written to MATVEL files.

ADPIC uses the model parameters from the ADPIC.NML file to release marker particles which simulate the actual release of effluent into the wind field. The grid cell concentrations for specified heights for specified plot types for a specified time interval are written to CONC\*.BIF files, corresponding CONC\*.VLST files are also written containing the MEDVEL/MATVEL filenames used in generating the CONC\*.BIF file. Information for individual particles can be saved in the ADPIC\*.RST file, so that an ADPIC run can be restarted. Information required for displaying dots over terrain is contained in PARTPOS\*.BIF files.

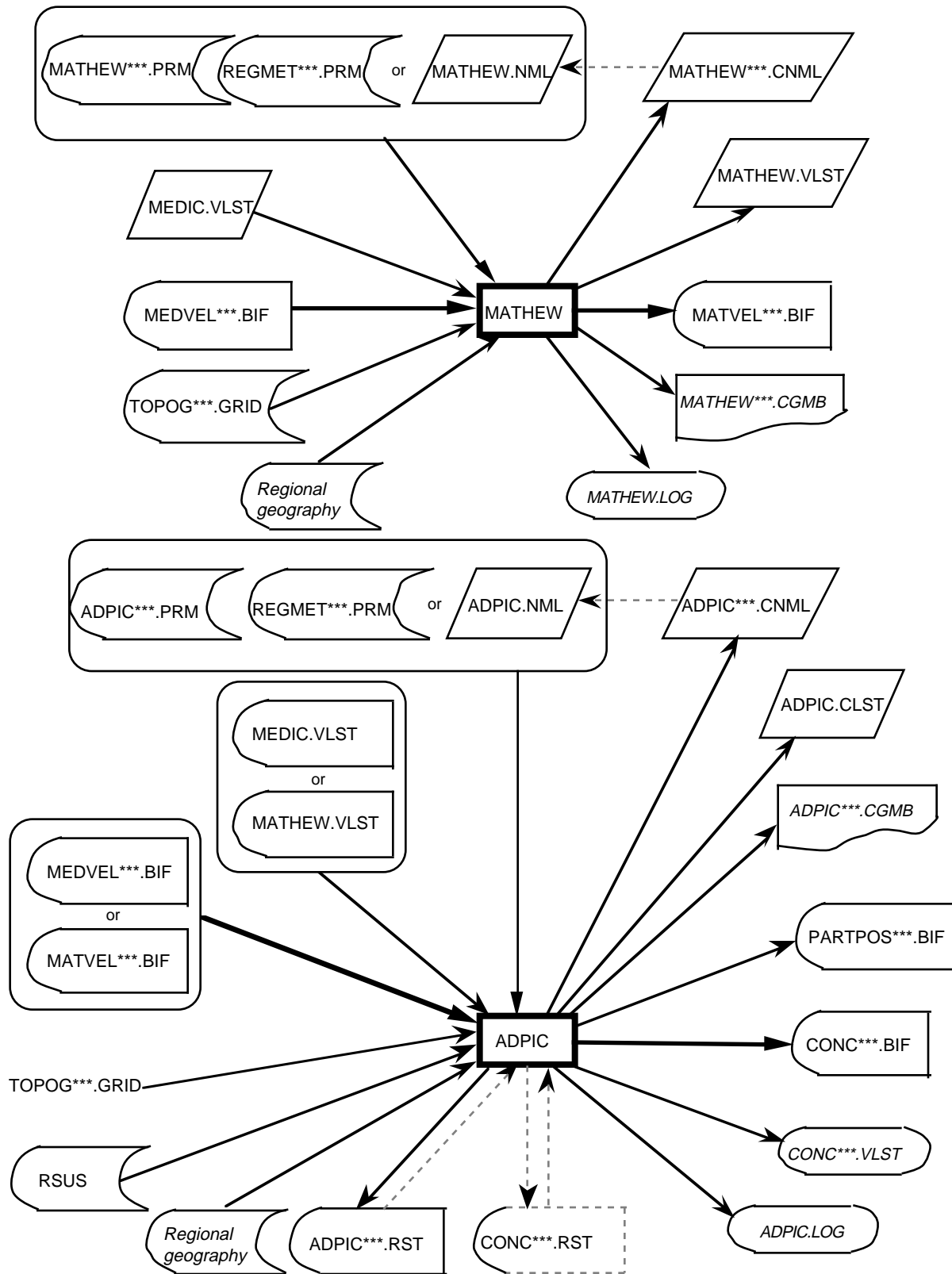
PLCNT and TIMEHIS are post-processor codes which use the parameters from the CONC\_ANALYSIS.NML file to display the concentration data from the CONC\*.BIF files in different ways: PLCNT generates contour plots of the concentration data, while TIMEHIS generates semi-log time history plots of the concentration data.



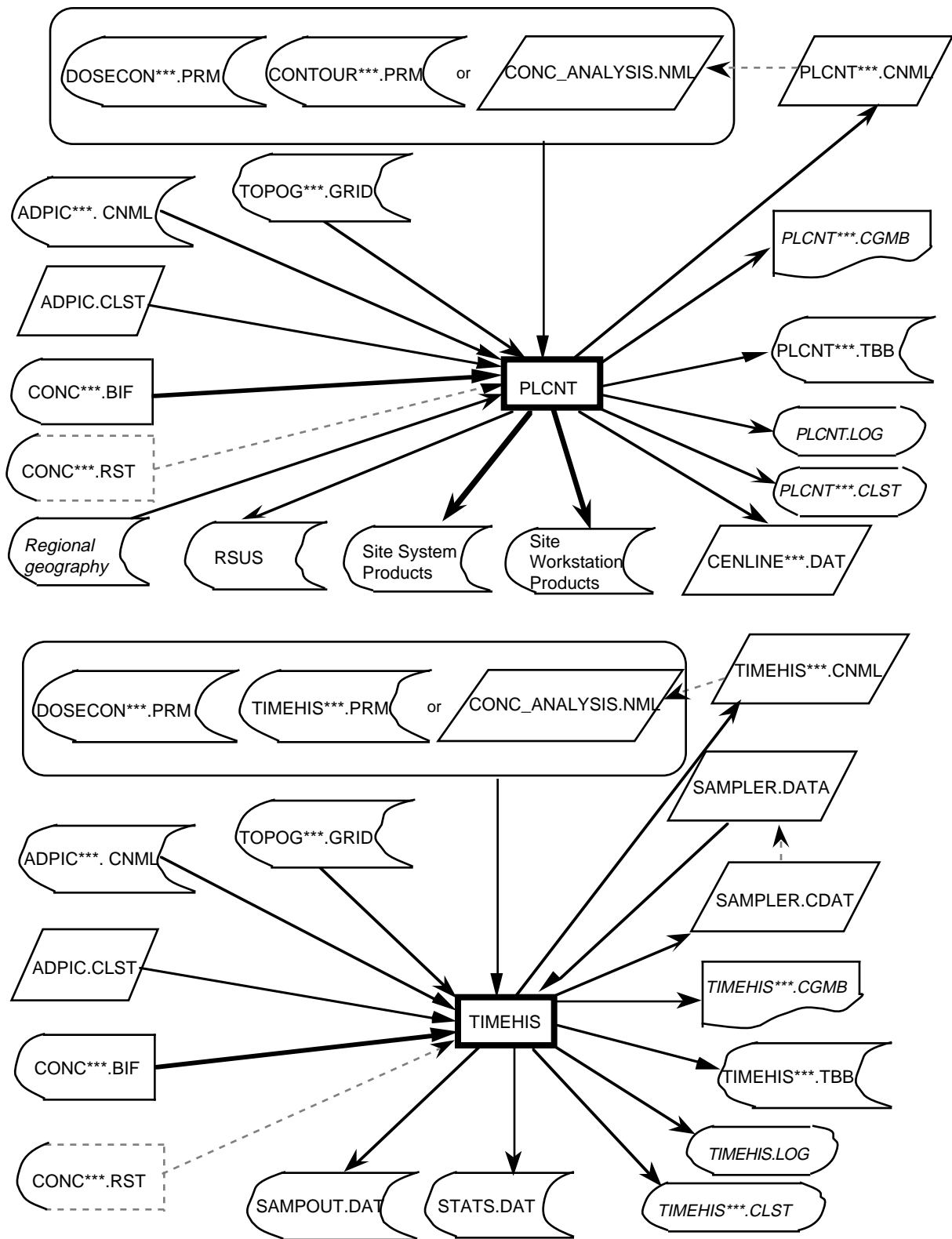
**Fig. III.A.1 Overview of the CG-MATHEW/ADPIC Models**



**Fig. III.A.2 Detail for TOPOG (top) and MEDIC (bottom)**



**Fig. III.A.3 Detail for CG-MATHEW (top) and ADPIC (bottom)**



**Fig. III.A.4 Detail for PLCNT (top) and TIMEHIS (bottom)**

### III.B. Input Files

There are several conventions that are followed in all the model input files. All program control flags are input as single characters with acceptable values of 'Y', 'y', 'N', and 'n'. Horizontal distances are input in kilometers and vertical distances in meters. Dates are all input as seven character strings in ddMMMy format where dd is the day of the month, MMM is the alphabetic month (case insensitive) and yy is the last two digits of the year. The date/time utilities in the models work from January 1, 1950 to December 31, 2049 by assuming the yy values of less than 50 are in the next century and yy values of greater than 50 are in this century. Times are strings of up to eight characters and can be in one of the four following formats:

'hhmm'

'hhmmss'

'hh:mm'

'hh:mm:ss'

where hh is the hours (<24), mm is the minutes (<60) and ss is an optional seconds field (<60). Note the in the description of the time input variables below, they will typically be described as being in 'hhmmss' format although all of the formats above are acceptable. There are also a number of time interval inputs (e.g., ...\_INTERVAL variables in ADPIC and ...\_STEP variables in PLCNT) where a prefix can be added to a legal time string to indicate a number of days (<1000) in the duration. Thus, the following is a legal interval input string (note that the days field along with the double colon are not required if the interval is less than 24 hours but if the days field is included then the double colon must separate the days and hours fields):

'DDD::hh:mm:ss'



### III.B1. TOPOG.NML

TOPOG.NML is the input file of grid size and position information for the TOPOG code. Since TOPOG creates a grid that is fixed in space, a large enough area must be chosen to account for the wind and stability conditions that might affect the effluent as the problem unfolds. An overview of the input is necessary because of the appreciable interplay between the individual parameters in TOPOG.NML as well as other parameters in the code system.

Since the TOPOG.NML input defines a three-dimensional grid for ADPIC, the first step in preparing the file is to estimate the area of interest for the duration of the problem. The overall objective is to keep at least 90% of the particles within the grid volume so as to maximize resolution of the effluent puff or plume. We have found streamline analyses using past, present and, if available, forecast weather maps to be extremely helpful in estimating the effluent trajectory. A calculated mean wind speed in conjunction with this directional information will then serve to guide the analyst in choosing the proper horizontal ( $x$  and  $y$ ) grid dimensions. Care should be taken to insure that the resulting ADPIC grid will have positive  $x$ - $y$  coordinates, since PLCNT will fail in low level graphics calls with negative coordinates.

In selecting the vertical grid dimension, it is necessary to consider the height of the release, topography, upper air winds, and the stability of the atmosphere. Topography is one of the more important considerations in this decision process. TOPOG assigns a zero height to the lowest terrain elevation in the grid. The top of the grid is generally chosen to be above the highest topographic feature. There are two exceptions to this rule: (1) if an isolated mountain peak extends well above the nearby topography, the peak should be allowed to protrude through the top of the grid so as to improve the resolution elsewhere, and (2) if it is highly unlikely that the effluent cloud would rise above a particular terrain level, it may be advantageous to lower the grid top to that level.

There are two ways of altering the grid dimension: (1) the size of the grid cell can be varied in the TOPOG.NML file as described below, and (2) the number of grid points along each axis can be varied. (The standard CG-MATHEW and ADPIC grids have 51 by 51 by 15 points and 41 by 41 by 15 points, respectively.) The first option is the one of choice in most ARAC problems. However, there may be times when it is desirable to change the number of grid points to achieve the optimum grid resolution for a given grid volume. In order to do this, the grid point parameters must be changed in appropriate include files and the codes recompiled. The parameter names in TOPOG, MEDIC, and CG-MATHEW and ADPIC are IMAXPIC, JMAXPIC, and KMAXPIC for the ADPIC grid and IMAXMAT, JMAXMAT, and KMAXMAT for the CG-MATHEW grid. (Computer memory and execution speeds are limiting factors in choosing a total number of grid points.)

The following is a list of the TOPOG.NML variables grouped alphabetically within namelist blocks:

## **II.B1a. TOPOG\_CONTROL namelist**

### **CONTOUR INTERVAL**

The model terrain contour interval in meters.

(Default: CONTOUR\_INTERVAL = DELZ)

Warning errors: CONTOUR\_INTERVAL  $\leq$  1.0

Fatal errors: CONTOUR\_INTERVAL  $\leq$  0.001

CONTOUR\_INTERVAL > 5000.0 (M/A)

### **FLAT TOPO**

Flag ('Y' or 'N') indicating if flat terrain is to be used. Special Note for HM/A: valid values for the terrain file (defined by the “Select Topography” menu option) are: NHEMISTD, NHEMIAVG, NHEMIMAX.

(Default: FLAT\_TOPO = 'N' → do not use flat terrain)

### **INTERACTIVE**

Flag ('Y' or 'N') specifying whether or not user interaction is required during the running of TOPOG. Minimal user interaction involves resolution of any 1\* DELX “holes”; in the noninteractive mode, the holes are filled automatically.

(Default for HANDSOFF mode: INTERACTIVE = 'N' Æ no user interaction; Default for non-HANDSOFF mode: INTERACTIVE = 'Y' Æ user interaction required)

### **LISTDETAILS**

Flag ('Y' or 'N') specifying whether or not additional information about the TOPOG grid will be written to the echo file. The additional information consists of: cell heights and DELZ heights.

(Default: LISTDETAILS = 'N' → no additional information is written to the echo file)

### **RESTART**

Flag ('Y' or 'N') indicating that an existing TOPOG\*.GRID file is to be opened and read. This allows a previous grid to be modified interactively without rerunning all of TOPOG.

(Default: RESTART = 'N' → start from scratch and create new TOPOG\*.GRID)

### **III.B.1.b. TOPOG\_PARAMS namelist**

#### **DELX**

The grid cell size in kilometers along the  $x$ -axis. There are 40 grid cells for ADPIC and 47 grid cells for HADPIC. DELX must be an integer multiple of one-quarter the terrain database resolution. For a standard ARAC terrain file for a M/A run, the resolution is 0.5km, so DELX must be a multiple of 0.125km. DELX must be 381.0 for HM/A.

Warning errors: none

Fatal errors: DELX  $\leq 0.001$   
DELX  $> 1250.0$

#### **DELY**

The grid cell size in kilometers along the  $y$ -axis. There are 40 grid cells for ADPIC and 51 grid cells for HADPIC. DELY must be an integer multiple of one-quarter the terrain database resolution. For a standard ARAC terrain file for an M/A run, the resolution is 0.5km, so DELY must be a multiple of 0.125km. DELY must be 381.0 for HM/A.

Warning errors: none

Fatal errors: DELX  $\leq 0.001$   
DELX  $> 1250.0$

#### **DELZ**

The grid cell size in meters along the  $z$ -axis. There are 14 grid cells (15 levels) in both M/A and HM/A.

Warning errors: DELZ  $> 1000.0$   
DELX  $< 1.0$

Fatal errors: DELZ  $\leq 0.001$   
DELZ  $> 5000.0$

#### **DOUBLECELL**

Flag ('Y' or 'N') specifying whether or not grid cell heights will be assigned in blocks of four grid cells from the lower left corner of the grid.

(Default: DOUBLECELL = 'N'  $\rightarrow$  blocks of four will not be used, i.e., the height for each grid cell is determined individually.)

#### **MAP\_ADJUSTMENT**

Flag ('Y','N') specifying whether or not a map projection adjustment is to be made which would align map north with true north for the map projection selected. One effect of setting this flag is to correct the wind direction arrays, and adjust the wind speed to account for local scale variations

caused by the map projection. Note: In order to use MAP\_ADJUSTMENT = 'Y', the map projection logicals need to be set, i.e., you may need to have selected a map projection.

(Default: MAP\_ADJUSTMENT = 'Y')

### **RANGE**

ADPIC grid extent along both the x- and y axes in kilometers. RANGE can be used to specify both X RANGE and Y RANGE. RANGE may also be used instead of DELX—also see notes for DELX.

Warning errors: RANGE < 1.0  
Fatal errors: RANGE ≤ 0.5  
RANGE > 45000.0

### **X GRIDCENTER**

The model x coordinate of the center of the ADPIC grid, input in kilometers.

Note: X\_GRIDCENTER and Y\_GRIDCENTER may be used instead of X\_ORIGIN\_PIC and Y\_ORIGIN\_PIC.  
Special note for HM/A: X\_GRIDCENTER = 8763.  
Fatal errors: X\_GRIDCENTER ≤ -45000.0  
X\_GRIDCENTER > 45000.0

### **X OFFSET MAT**

Real number specifying the offset (km) in the x-direction of the ADPIC grid from the CG-MATHEW grid. The ADPIC grid must be contained in the CG-MATHEW grid; therefore X\_OFFSET\_MAT must be less than 0.5 \* (CG-MATHEW grid size - ADPIC grid size).

(Default: X\_OFFSET\_MAT = 0.0 → no offset)

### **X ORIGIN PIC**

The Model x coordinate of the lower left-hand corner of the ADPIC grid in kilometers.

Note: X\_ORIGIN\_PIC and Y\_ORIGIN\_PIC may be used instead of X\_GRIDCENTER and Y\_GRIDCENTER.  
Fatal errors: X\_ORIGIN\_PIC ≤ -45000.0  
X\_ORIGIN\_PIC > 45000.0

### **XRANGE**

ADPIC grid extent along the x-axis in kilometers. XRANGE may be used instead of DELX—also see notes for DELX.

Warning errors: XRANGE < 1.0  
Fatal errors: XRANGE ≤ 0.5  
XRANGE > 45000.0

### **Y GRIDCENTER**

The model y coordinate of the center of the ADPIC grid, input in kilometers.

Note: X\_GRIDCENTER and Y\_GRIDCENTER may be used instead of X\_ORIGIN\_PIC and Y\_ORIGIN\_PIC.

Special note for HM/A: Y\_GRIDCENTER = 9525.,

Fatal errors: Y\_GRIDCENTER  $\leq$  -45000.0

Y\_GRIDCENTER > 45000.0

### **Y OFFSET MAT**

Real number specifying the offset (km) in the y-direction of the ADPIC grid from the CG-MATHEW grid. The ADPIC grid must be contained in the CG-MATHEW grid; therefore Y\_OFFSET\_MAT must be less than 0.5 \* (CG-MATHEW grid size - ADPIC grid size).

(Default: Y\_OFFSET\_MAT = 0.0  $\rightarrow$  no offset)

### **Y ORIGIN PIC**

The Model y coordinate of the lower left-hand corner of the ADPIC grid in kilometers.

Note: X\_ORIGIN\_PIC and Y\_ORIGIN\_PIC may be used instead of

X\_GRIDCENTER and Y\_GRIDCENTER.

Fatal errors: Y\_ORIGIN\_PIC  $\leq$  -45000.0

Y\_ORIGIN\_PIC > 45000.0

### **YRANGE**

ADPIC grid extent along the y-axis in kilometers. YRANGE may be used instead of DELY–also see notes for DELY.

Warning errors: YRANGE < 1.0

Fatal errors: YRANGE  $\leq$  0.5

YRANGE > 45000.0

### **ZRANGE**

ADPIC grid extent along the z-axis in meters. ZRANGE may be used instead of DELZ – also see notes for DELZ.

Warning errors: ZRANGE < 10.0 (M/A)

ZRANGE  $\geq$  5000.0

Fatal errors: ZRANGE > 70000.0 (M/A)

### Sample TOPOG.NML Input File (with center reference)

\$stopin_control			beginning of control namelist
FLAT_TOPO	= 'Y'		flag for using flat terrain
RESTART	= 'N'		flag for using existing GRIDIN file
INTERACTIVE	= 'Y'		flag for non-interactive execution
LISTDETAILS	= 'N'		flag for writing additional information to echo file
Send			end of control namelist
\$stopin_params			beginning of params namelist
X_GRIDCENTER	= 115.0	←	X of center of ADPIC model grid
Y_GRIDCENTER	= 115.0	←	Y of center of ADPIC model grid
RANGE	= 30.000	←	range (km) of ADPIC grid in x- and y-directions
ZRANGE	= 700.	←	range (m) of ADPIC grid in z-direction
X_OFFSET_MAT	= 0.000	←	offset (km) for CG-MATHEW grid in x-direction
Y_OFFSET_MAT	= 0.000	←	offset (km) for CG-MATHEW grid in y-direction
DOUBLECELL	= 'N'	←	flag for assigning grid cell heights in blocks of four
Send			end of params namelist

### Sample TOPOG.NML Input File (with lower left corner reference)

\$stopin_control			beginning of control namelist
FLAT_TOPO	= 'Y'		flag for using flat terrain
RESTART	= 'N'		flag for using existing GRIDIN file
INTERACTIVE	= 'Y'		flag for non-interactive execution
LISTDETAILS	= 'N'		flag for writing additional information to echo file
Send			end of control namelist
\$stopin_params			beginning of params namelist
X_ORIGIN_PIC	= 100.0	←	X of lower left corner of ADPIC model grid
Y_ORIGIN_PIC	= 100.0	←	Y of lower left corner of ADPIC model grid
RANGE	= 30.000	←	range (km) of ADPIC grid in x- and y-directions
ZRANGE	= 700.	←	range (m) of ADPIC grid in z-direction
X_OFFSET_MAT	= 0.000	←	offset (km) for CG-MATHEW grid in x-direction
Y_OFFSET_MAT	= 0.000	←	offset (km) for CG-MATHEW grid in y-direction
DOUBLECELL	= 'N'	←	flag for assigning grid cell heights in blocks of four
Send			end of params namelist

### III.B.2. Geography

The geography file contains location and classification information about geographic features (e.g. roads, lakes, rivers, political boundaries). This information is used to plot a base map for some of the model graphics products. These files are normally not provided with the models.

### III.B.3. Regional Elevation Data

The regional elevation file contains the altitudes of the topographic surface around a site. It is composed of a header which describes the grid being used, followed by a two-dimensional array of cell heights, one for each grid cell. The header used in this file is used in exactly the same format for other types of geographic data in the ARAC system in addition to the elevation data described here. In the ARAC system the elevation data is extracted from an elevation data base covering much of the world that has been designed for ARAC applications. The regional elevation file is a binary file consisting of the following data:

Variable	Dimension	Type	Description
MAP_FILENAME	1	CHAR*32	Map name
DESCRIPTION	1	CHAR*80	Explanatory text
DATA_TYPE	1	CHAR*20	Indicates type of data. e.g., elevations.
PROJECTION_TYPE	1	CHAR*32	String indicating the map projection for the data (see the <i>ARAC Map Projection System</i> for details on this and other projection parameters)
PROJECTION_CENTRAL_LON	1	REAL*4	Central longitude of projection in decimal degrees (ddd.dddd)
PROJECTION_CENTRAL_LAT	1	REAL*4	Central latitude of projection in decimal degrees (ddd.dddd)
PROJECTION_HEMI	1	CHAR*4	Latitude hemisphere indicator ('N'/'S'), not normally used.
PROJECTION_STD_PAR	2	REAL*4	Standard parallels for the projection
PROJECTION_ELLIPSOID	1	INTEGER*4	Indicator of earth figure (sphere or ellipsoid)
PROJECTION_DATUM	1	INTEGER*4	Indicator of datum for this data.
LIMIT_EAST	1	REAL*4	Maximum longitude of region covered by this file in decimal degrees (ddd.dddd).
LIMIT_WEST	1	REAL*4	Minimum longitude of region covered by this file in decimal degrees (ddd.dddd).
LIMIT_NORTH	1	REAL*4	Maximum latitude of region covered by this file in decimal degrees (ddd.dddd).
LIMIT_SOUTH	1	REAL*4	Minimum latitude of region covered by this file in decimal degrees (ddd.dddd).
X_MINIMUM	1	REAL*4	Minimum X of region covered by this file.
X_MAXIMUM	1	REAL*4	Maximum X of region covered by this file.
Y_MINIMUM	1	REAL*4	Minimum Y of region covered by this file.
Y_MAXIMUM	1	REAL*4	Maximum Y of region covered by this file.
LATITUDE_OF_INTEREST	1	REAL*4	Latitude of important point in file (usually the center) in decimal degrees (ddd.dddd)

LONGITUDE_OF_INTEREST	1	REAL*4	Longitude of important point in file (usually the center) in decimal degrees (ddd.dddd)
X_OF_INTEREST	1	REAL*4	X of important point in file (usually the center) in decimal degrees (ddd.dddd)
Y_OF_INTEREST	1	REAL*4	Y of important point in file (usually the center) in decimal degrees (ddd.dddd)
X_CELL_SIZE	1	REAL*4	Grid cell size in the x direction for raster geographic data, currently this is used only for elevation data.
Y_CELL_SIZE	1	REAL*4	Grid cell size in the y direction for raster geographic data, currently this is used only for elevation data.
X_DIMENSION	1	INTEGER*4	Dimension in x direction of raster geographic data, currently used only for elevation data.
Y_DIMENSION	1	INTEGER*4	Dimension in y direction of raster geographic data, currently used only for elevation data.
ELEVATION_MINIMUM	1	INTEGER*4	Lowest elevation in an elevation data file.
ELEVATION_MAXIMUM	1	INTEGER*4	Highest elevation in an elevation data file.
DATA_SOURCE	1	CHAR*20	Source from which this data file was derived e.g., DMA DTED, DMA planar, USGS 1:2,000,000
VERSION	1	INTEGER*4	Version number for editable files
HEADER_PAD	56	INTEGER*4	Header pad fills to a 512 byte block
ELEVATION_GRID	400 by 400 (default)	INTEGER*2	Geographic data. For input into TOPOG only regional elevation data files are currently used.

---

## **ELEVATION GRID**

The array of elevations is based on the grid defined in the header. Each grid cell is associated with an array index pair  $(i, j)$  with the pair  $(1, 1)$  indicating the grid cell at the southwest corner of the grid. The value of an array element is the average height in meters above sea level of the corresponding grid cell. An array value of -9999 indicates that no valid data was available to generate a value for this grid cell. TOPOG treats these missing data flags as sea level (0) heights. The standard ARAC terrain data base is a 400 by 400 array; thus, there are 160,000 height values stored as 2-byte integers in a standard regional elevation file.

### **III.B.4. MEDIC.NML**

MEDIC.NML is the principal input file for control of the MEDIC code. It provides MEDIC with numerous parameters controlling the wind field extrapolation or interpolation procedures. Two types of Fortran namelists make up the MEDIC.NML file. The first, MEDIC\_CONTROL, permits the user to control general processing and output options. The second, MEDIC\_PARAMS, controls parameters which are specific to a set of meteorological data. A separate MEDVEL output file is created for each occurrence of a MEDIC\_PARAMS namelist within MEDIC.NML. These MEDVEL data sets are used by CG-MATHEW to create snapshots of the atmospheric winds adjusted for mass-consistency throughout a gridded volume.



Meteorological data may come from an OBSERV.MET file or Problem Metdata files, and/or GWC or RWM gridded met data files. The variable METDATASET given in a MEDIC\_PARAMS namelist is matched to a keyword which prefices every METDATASET within an OBSERV.MET file. Typically, the MEDIC\_PARAMS namelist and the corresponding METDATASET within the OBSERV.MET file track at intervals which match data availability, typically every hour.

Because CG-MATHEW does a minimal wind field adjustment and because the advection winds in ADPIC persist for extended periods, it is very important that the analyst provide the best possible set of extrapolated winds. The extrapolated winds are sensitive to relatively small changes in the MEDIC parameter values, and the effects on the wind field are often complex and hard to visualize. It is therefore essential that the assessor (a meteorologist) carefully examines the plots of the extrapolated winds which MEDIC produces to determine the validity of the wind field. If the wind field fits the assessor's concept of the state of the atmosphere, then CG-MATHEW may be run with the data set. Otherwise, several of the variables in MEDIC.NML may have to be changed and MEDIC run again. It may also be necessary to omit meteorological data if it is obviously erroneous, e.g., several hours of unchanged direction readings from a station may indicate a "frozen" wind vane. This process is iterative, and is continued until the assessor is satisfied that the best extrapolation (consistent with the data) has been produced. If only gridded upper air data is used, profile extrapolation is generally to be preferred over parameterized extrapolation (PROFILE\_EXTRAP = 'Y').

The hemispheric variant HMEDIC (see Part II.C.2.e) uses a subset of the MEDIC.NML namelist variables. All of the MEDIC\_CONTROL namelist variables apply *except* for REGMET\_PARAMFILE, SFC\_ROUGH\_HGT, and STNLOCPSL. However, since HMEDIC uses only AFGWC gridded winds and does not perform extrapolation, *only* MET\_START\_DATE, MET\_START\_TIME, AVGTIME, NHEDATA, NHEFCST, NHETIME, SHEDATE, SHEFCST, SHETIME are used from the MEDIC\_PARAMS namelist.

The following is a list of variables for the two MEDIC namelists, including defaults. Variables not marked as (MEDIC/HMEDIC variable) apply to MEDIC *only*.

### **III.B.a. MEDIC\_CONTROL namelist**

#### **GWC BARB PLOT FREQ** (MEDIC/HMEDIC variable)

Integer value specifying the density of barbs on the GWC wind barb plots. A value of one (1) indicates that a vector is plotted for every observation. A value of two (2) indicates that a vector is plotted for every second observation, etc.

(Default : 1)

Warning errors: None

Fatal errors: GWC\_BARB\_PLOT\_FREQ <= 0

### **GWC BARB SCALE** (MEDIC/HMEDIC variable)

Floating point value scaling the size of the barbs on the GWC wind barb plots in the horizontal direction.

(Defaults : GWC\_BARB\_SCALE = 0.35 for 100< # stns <500,

GWC\_BARB\_SCALE = 0.6 for # stns <100,

GWC\_BARB\_SCALE = 0.2 for #stns >500)

Warning errors: None

Fatal errors: GWC\_BARB\_SCALE < = 0.0

### **HORZ VECTOR SCALE** (MEDIC/HMEDIC variable)

Floating point value which scales the size of the vectors on the vector plots in the horizontal direction.

(Default : 0.5 for MEDIC, 0.1 for HMEDIC)

Warning errors: None

Fatal errors: HORZ\_VECTOR\_SCALE < = 0.0

### **IPROFILE** (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard CG-MATHEW grid) representing the *i* indices of the sample profiles to be plotted. The first value in IPROFILE is matched with the first value in JPROFILE to give the horizontal grid indices of the first profile. Succeeding indices from each list are matched in a similar fashion. The default for IPROFILE with a standard 51 by 51 by 15 CG-MATHEW grid is: 14 26 38 14 26 38 14 26 38. This gives the normal nine sample profiles. The variables XPROFILE and YPROFILE allow desired locations for vertical profiles to be entered in current map coordinates. See the description of those variables for more information.

### **JPROFILE** (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard CG-MATHEW grid) representing the *j* indices of the sample profiles to be plotted. The first value in JPROFILE is matched with the first value in IPROFILE to give the horizontal grid indices of the first profile. Succeeding values from each list are matched in a similar fashion. The default for JPROFILE with a standard 51 by 51 by 15 CG-MATHEW grid is: 14 14 14 26 26 26 38 38 38. This gives the normal nine sample profiles. The variables XPROFILE and YPROFILE allow desired locations for vertical profiles to be entered in current map coordinates. See the description of those variables for more information.

**MEDIC\_VLST** (MEDIC/HMEDIC variable)

6-character string ('NEW', 'NONE', 'MODIFY') which controls the creation of the MEDIC.VLST file. If set to 'NEW', a MEDIC.VLST file is created containing a list of the fully qualified MEDVEL file names produced by the run. If set to 'NONE', no MEDIC.VLST file is created. If the flag is set to 'MODIFY', the existing MEDIC.VLST file is read in and updated from the MEDVEL files created during the current run.

(Default MEDIC\_VLST = 'NEW'.)

Warning errors: None

Fatal errors: None

**NODDS BARB PLOT FREQ** (MEDIC/HMEDIC variable)

Integer value (1,2,?) controlling the barb density: one out of every NODDS\_BARB\_PLOT\_FREQ will be plotted on the NODDS horizontal vector plots.

(Default NODDS\_BARB\_PLOT\_FREQ = 1 -> every barb is plotted)

**NODDS BARB SCALE** (MEDIC/HMEDIC variable)

Floating point value (>0, usually less than 2) that scales the barbs on the NODDS horizontal vector plots.

(Default NODDS\_BARB\_SCALE = 1 -> default barb size is used)

**PLOT LEVELS** (MEDIC/HMEDIC variable)

Array (max size = KMAXMAT = 15) of integers (1 to 15 for standard MEDIC grid) representing the *K* levels above terrain at which horizontal vector plots are to be produced.

(Default PLOT\_LEVELS : 1, 2, 7, 15)

**PLOT LEVELS NHE** (MEDIC/HMEDIC variable)

Array (max size = KMAX\_UPR\_HEMI = 15 currently, the number of hemispheric grid levels) of integers representing the Northern Hemisphere data grid levels to be plotted in the MEDIC graphics file. For example, PLOT\_LEVELS\_NHE=1 4 15 will plot the first (1000~mb), fourth (500~mb), and fifteenth (10~mb) levels.

(Default PLOT\_LEVELS\_NHE : 1, 5, 10, 15)

**PLOT LEVELS NODDS** (MEDIC/HMEDIC variable)

Array (max size = KMAX\_UPR\_NODDS = 9) of integers representing the NODDS (Navy Oceanographic Data Display System) data grid levels (1000mb, 925mb, 850mb, 700mb, 500mb, 400mb, 300mb, 200mb, 100mb) to be plotted as barbs in the MEDIC graphics file. For example, PLOT\_LEVELS\_NODDS=1 2 3 5 will plot the first (1000mb), second (925mb), third (850mb), and fifth (500mb) levels.

(Default PLOT\_LEVELS\_NODDS = 0 -> no levels plotted)

### **PLOT\_LEVELS\_RWM**

Array (max size = KMAX\_UPR\_RWM = 9 currently, the number of Relocatable Window Model levels) of integers representing the Relocatable Window Model grid levels to be plotted in the MEDIC graphics file. For example, PLOT\_LEVELS\_RWM = 1 2 6 9 will plot the first (surface), second (1000~ft), sixth (15000~ft), and ninth (30000~ft) levels.

### **PLOT\_LEVELS\_SHE** (MEDIC/HMEDIC variable)

Array (max size = KMAX\_UPR\_HEMI = 15 currently, the number of hemispheric grid levels) of integers representing the Southern Hemisphere data grid levels to be plotted in the MEDIC graphics file. For example, PLOT\_LEVELS\_SHE= 1 4 15 will plot the first (1000~mb), fourth (500~mb), and fifteenth (10~mb) levels.

(Default PLOT\_LEVELS\_SHE : 1, 2, 7, 15)

### **PRINT\_WINDFIELDS** (MEDIC/HMEDIC variable)

Flag ('Y', 'N') that controls if the reference level and upper level winds (MEDIC) or the GWC input winds (HMEDIC) and the extrapolated wind fields are to be printed.

(Default : 'N' -> none of the above wind fields are printed)

Warning errors: None

Fatal errors: None

### **REGMET\_PARAMFILE**

22-character string which gives the name of the regmet parameter file to be used. The default value, ' ', is used to check if a string was given in the MEDIC\_CONTROL namelist. (WARNING: Not yet implemented! !)

(Default : ' ')

Warning errors: None

Fatal errors: None

### **RWM\_BARB\_PLOT\_FREQ** (MEDIC/HMEDIC variable)

Integer value that controls the density of barbs on the RWM wind barb plots. A value of one (1) indicates that a vector is plotted for every observation. A value of two (2) indicates that a vector is plotted for every second observation, etc.

(Default : 1)

Warning errors: None

Fatal errors: RWM\_BARB\_PLOT\_FREQ <= 0

**RWM BARB SCALE** (MEDIC/HMEDIC variable)

Floating point value that scales the size of the barbs on the RWM wind barb plots in the horizontal direction.

(Default : set according to the number of barbs plotted)

Warning errors: None

Fatal errors: RWM\_BARB\_SCALE < = 0.0

**SFC BARB PLOT FREQ**

Integer value that controls the density of barbs on the surface wind barb plots. A value of one (1) indicates that a vector is plotted for every surface observation. A value of two (2) indicates that a vector is plotted for every second surface observation, etc.

(Default : 1)

Warning errors: None

Fatal errors: SFC\_BARB\_PLOT\_FREQ < = 0

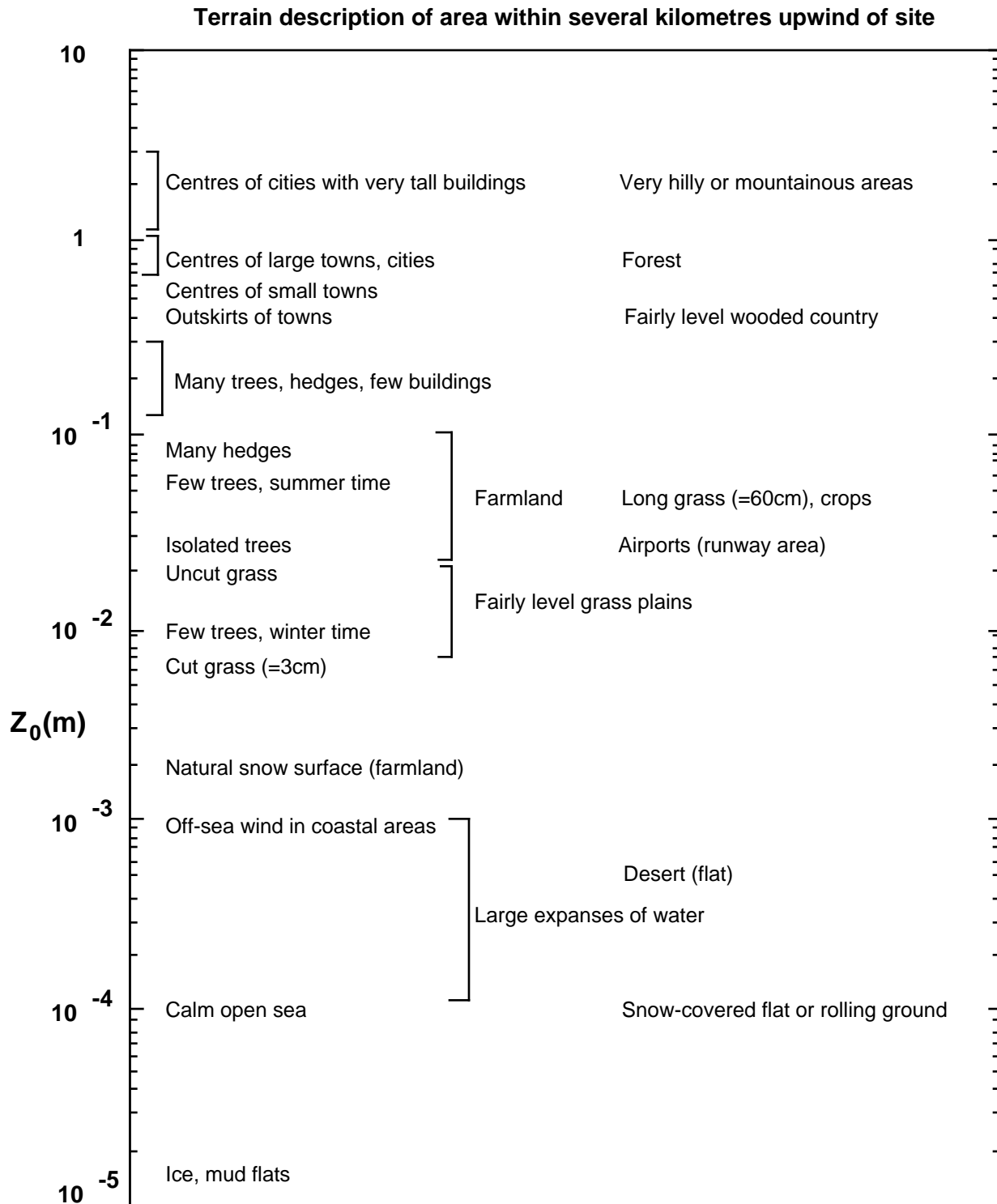
**SFC BARB SCALE**

Floating point value that scales the size of the barbs on the surface wind barb plots in the horizontal direction.

(Default : set according to the number of barbs plotted)

Warning errors: None

Fatal errors: SFC\_BARB\_SCALE < = 0.0



**Figure III.B.3.  $z_0$  values for typical terrain types (After ESDU 72026, 1972)**

## **SFC ROUGH HGT**

Floating point value that represents the surface roughness height in meters. The surface roughness height is approximately the average height of most land-based features (e.g., grass, trees, buildings) divided by 30. The analyst should consider features over the entire gridded area before entering a value. This is a subjective parameter since we are usually dealing with very diverse geographical areas. Figure III.B.3 gives typical values of surface roughness for various terrain types.

Warning errors: SFC\_ROUGH\_HGT  $\geq$  2.0

Fatal errors: SFC\_ROUGH\_HGT  $\leq$  0.0

SFC\_ROUGH\_HGT  $>$  5.0

## **STNLOCPSL**

Flag ('Y', 'N') indicating whether or not to use the problem station library.

(Default: 'N' -> either no station location data is required or it will be provided in STNLOC.MET)

## **TERRAIN ADJUSTMENT** (MEDIC/HMEDIC variable)

Flag ('Y', 'N') specifying whether the winds should be terrain shifted ('Y') or zeroed inside the terrain ('N').

(Default : 'Y' for MEDIC and 'N' for HMEDIC)

## **VECTOR PLOT FREQ** (MEDIC/HMEDIC variable)

Integer value controlling the density of vectors on the vector plots. A value of one (1) indicates that a vector is plotted at every horizontal grid point. A value of two (2) indicates that a vector is plotted at every other grid point, etc. (Default : VECTOR\_PLOT\_FREQ = 2)

Warning errors: None

Fatal errors: VECTOR\_PLOT\_FREQ  $\leq$  0

## **XPROFILE** (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of floating point values representing the x-coordinate of the sample profile locations. XPROFILE/YPROFILE constitute a list of the map coordinates of the sample profiles to be plotted. The first value in XPROFILE is matched with the first value in YPROFILE to give the horizontal grid indices of the first profile. Succeeding values from each list are matched in a similar fashion. The XPROFILE/YPROFILE pair is interpolated to the nearest horizontal grid location. The map coordinates in arrays XPROFILE/YPROFILE will in all instances override the values in IPROFILE/JPROFILE, including defaults. If no values are entered for XPROFILE/YPROFILE, the values in IPROFILE/JPROFILE are used.

## **YPROFILE** (MEDIC/HMEDIC variable)

Array (max size = MAXPROF = 10) of floating point values representing the y-coordinate of the sample profile locations. See XPROFILE for a full description.

### **III.B.4 .b MEDIC\_PARAMS namelist**

#### **ALL\_WINDS\_DIR** (MEDIC/HMEDIC variable)

Array (max size = MAXLEVEL\_ALL) of floating point values representing the wind direction profile for a uniform wind field. Uniform wind fields are created if the direction (ALL\_WINDS\_DIR), speed (ALL\_WINDS\_SPD), and height (ALL\_WINDS\_HGT) are set. No met station information (i.e., problem station library or STNLOC.MET) is required. Constant winds relative to the grid may be created by setting the TOPOG\_PARAMS namelist parameter MAP\_ADJUSTMENT = 'N'. Otherwise constant winds relative to geographic coordinates are assumed (MAP\_ADJUSTMENT = 'Y'). NOTE: In order to achieve circumpolar winds in HMEDIC, you can set ALL\_WINDS\_SPD to the speed at 60 deg latitude and leave MAP\_ADJUSTMENT = 'Y'; ALL\_WINDS\_DIR has no influence in this case. IMPORTANT: SINGLE\_PROFILE\_WINDS must be set to 'Y' to utilize ALL\_WINDS\_DIR.

#### **ALL\_WINDS\_HGT** (MEDIC/HMEDIC variable)

Array (max size = MAXLEVEL\_ALL) of floating point values representing the heights (m) above terrain of the levels specified in the ALL\_WINDS\_SPD and ALL\_WINDS\_DIR arrays. IMPORTANT: SINGLE\_PROFILE\_WINDS must be set to 'Y' to utilize ALL\_WINDS\_HGT.

#### **ALL\_WINDS\_SPD** (MEDIC/HMEDIC variable)

Array (max size = MAXLEVEL\_ALL) of floating point values representing the wind speed (m/s) profile for a uniform wind field. Uniform wind fields are created if the direction (ALL\_WINDS\_DIR), speed (ALL\_WINDS\_SPD), and height (ALL\_WINDS\_HGT) are set. No met station information (i.e., problem station library or STNLOC.MET) is required. Constant winds relative to the grid may be created by setting The TOPOG\_PARAMS namelist parameter MAP\_ADJUSTMENT = 'N'. Otherwise constant winds relative to geographic coordinates are assumed (MAP\_ADJUSTMENT = 'Y'). NOTE: In order to achieve circumpolar winds in HMEDIC, you can set ALL\_WINDS\_SPD to the speed at 60 deg latitude and leave MAP\_ADJUSTMENT = 'Y'; ALL\_WINDS\_DIR has no influence in this case. IMPORTANT: SINGLE\_PROFILE\_WINDS must be set to 'Y' to utilize ALL\_WINDS\_SPD.

#### **AVGTIME** (MEDIC/HMEDIC variable)

13-character string representing the time interval (DDD::hh:mm:ss) which is used to specify the averaging time applicable to a given dataset. This value is not used in the calculation, but is used to construct the fully qualified name of the resulting MEDVEL/MATVEL files. It is a required input.

#### **BL\_HGT**

Floating point value representing the height (m) of the boundary layer above terrain. The height of the boundary layer is the level at which the wind speed and direction become nearly geostrophic, i.e., change little with height because of the negligible effects of surface drag. Upper air sounding (RAWINS or PIBALS) are the best data source for determining BL\_HGT. If no soundings are



available, an alternative procedure to obtain BL\_HGT is to use Figure III.B.1. BL\_HGT as used by the model is the height at which the extrapolated upper air profile matches the observed upper air profile. The wind speed and direction above BL\_HGT matches the observed upper air profile if PROFILE\_EXTRAP='Y', but remains constant if PROFILE\_EXTRAP='N'.

Warning errors: BL\_HGT  $\leq$  50.0  
BL\_HGT  $\geq$  3000.0  
Fatal errors: BL\_HGT  $\leq$  0.0  
BL\_HGT  $\geq$  5000.0

### **EXTRAP EXPNT BL**

Floating point value representing the interpolation exponent for the boundary layer. It is used to interpolate winds within the boundary layer, and is valid between the top of the surface layer (SL\_HGT) and the top of the boundary layer (BL\_HGT). A more complete discussion of this very important parameter is contained in Section II.C.2.b.

Warning errors: EXTRAP\_EXPNT\_BL > 10.0  
Fatal errors: EXTRAP\_EXPNT\_BL < 0.0  
EXTRAP\_EXPNT\_BL > 50.0

### **MAX VEER PARAM VERT**

Integer value controlling the sense of the vertical direction extrapolation in the boundary layer, and is only used when PROFILE\_EXTRAP = 'N' (the parameterized extrapolation). It is the largest angle, measured in degrees, which will be resolved by veering (i.e., the winds rotate in a clockwise direction with height) rather than by backing.

(Default : 240)

Warning errors: NONE  
Fatal errors: MAX\_VEER\_PARAM\_VERT < 0  
MAX\_VEER\_PARAM\_VERT > 360

### **MAX VEER PROF HORZ**

Integer value controlling the sense of how upper air and surface direction discrepancies are resolved at the top of the surface layer, and is only used when PROFILE\_EXTRAP='N' (the profile extrapolation).

(Default : 180)

Warning errors: NONE  
Fatal errors: MAX\_VEER\_PROF\_HORZ < 0  
MAX\_VEER\_PROF\_HORZ > 360

### **METDATASET**

40-character string designating which met data set in the OBSERV.MET file is to be used for this case (i.e., with this MEDIC\_PARAMS namelist). The string must match EXACTLY that in the

file OBSERV.MET. Suggested string composition is 'yymmddhhq' for most cases. A given dataset may be used more than once, or not at all, in a given MEDIC run: OBSERV.MET is reprocessed for every MEDIC\_PARAMS namelist processed so that any dataset can be located. NOTE: This parameter should not be used simultaneously with any of the PROBMET\_DST... parameters; i.e., a run may not access metdata from both a problem metdata file and a OBSERV.MET file..

**MET\_START\_DATE** (MEDIC/HMEDIC variable)

7-character string indicating the day (ddmmmyy) for which a MEDVEL wind dataset is to be created.

**MET\_START\_TIME** (MEDIC/HMEDIC variable)

8-character string representing the time (hh:mm:ss) for which a MEDVEL wind dataset is to be created.

Warning errors: None

Fatal errors: MET\_START\_TIME < 0  
MET\_START\_TIME > 2400

**NHEDATE** (MEDIC/HMEDIC variable)

7-character string representing the date (ddmmmyy) of the Northern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**NHEFCST** (MEDIC/HMEDIC variable)

Integer specifying the forecast period (hh) of the Northern Hemisphere gridded metdata to be used by MEDIC. For example, a value of 00 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the for the hemispheric gridded data are 00, 12, 24, 36, 48, and 60. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**NHETIME** (MEDIC/HMEDIC variable)

8-character string specifying the valid time (hh:mm:ss) of the Northern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**NODDSDATE** (MEDIC/HMEDIC variable)

7-character string representing the date (ddmmmyy) of the NODDS gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**NODDSFCST** (MEDIC/HMEDIC variable)

Integer specifying the forecast period (hh) of the NODDS gridded metdata to be used by MEDIC. For example, a value of -1 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the for the NODDS gridded data are -1, 12, 24, 36, 48, 60, 72, 84, 96, 108, and 120. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**NODDSTIME** (MEDIC/HMEDIC variable)

8-character string specifying the valid time (hh:mm:ss) of the NODDS gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**PROBMET\_DST**

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata Files (surface, tower, and upper air) if the three Data Set Times are not explicitly given. PROBMET\_DST is used as the default for a missing Data Set Time if any, but not all, of PROBMET\_DST\_SFC, PROBMET\_DST\_UPR, or PROBMET\_DST\_TOW are specified. NOTE: This parameter should not be used simultaneously with the METDATASET parameter; i.e., a run may not access metdata from both a problem metdata file and a OBSERV.MET file..

**PROBMET\_DST\_SFC**

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata File for surface data. PROBMET\_DST\_SFC will override PROBMET\_DST for surface datasets. NOTE: This parameter should not be used simultaneously with the METDATASET parameter; i.e., a run may not access metdata from both a problem metdata file and a OBSERV.MET file.

**PROBMET\_DST\_TOW**

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata Files for tower data. PROBMET\_DST\_TOW will override PROBMET\_DST for tower datasets. NOTE: This parameter should not be used simultaneously with the METDATASET parameter; i.e., a run may not access metdata from both a problem metdata file and a OBSERV.MET file..

**PROBMET\_DST\_UPR**

9-character (yymmddhhq) string representing the Data Set Time of the Problem Metdata Files for upper air data. PROBMET\_DST\_UPR will override PROBMET\_DST for upper air datasets.. NOTE: This parameter should not be used simultaneously with the METDATASET parameter.; i.e., a run may not access metdata from both a problem metdata file and a OBSERV.MET file.

## **PROFILE EXTRAP**

Flag ('Y', 'N') specifying if upper air profiles are used to perform the vertical extrapolation. If PROFILE\_EXTRAP is 'N', only the extrapolation parameters are used to generate three-dimensional gridded winds from two-dimensional grids of winds. (Also, see BL\_HGT concerning the relationship between PROFILE\_EXTRAP and BL\_HGT.) If gridded upper air data is used, profile extrapolation is generally preferred.

(Default : 'N', => use parameterized profile method)

## **PWR LAW EXPNT SL**

Floating point value representing the power law exponent for the surface layer. It can be calculated directly from surface-layer wind observations in the vertical using the power law formula given by Eq. (C.1). Alternatively, it can be obtained as a function of surface roughness height (surface roughness height is approximately 1/30th of the average height of features such as buildings, grass, trees) as shown in Fig. III.B.1, and the following correction for atmospheric stability applied:

Stable:	add 0.1
Neutral:	as is
Unstable:	subtract 0.1

PWR\_LAW\_EXPNT\_SL is used within the code to extrapolate: (1) wind speeds from their measured height to REF\_HGT (see below), (2) wind speeds from the SFC\_ROUGH\_HGT to the top of the surface layer above terrain in the parameterized extrapolation procedure (see SL\_HGT below), and (3) wind speeds from the SFC\_ROUGH\_HGT and REF\_HGT in the profile extrapolation procedure (see Section II.C.2.b for details).

Warning errors: PWR\_LAW\_EXPNT\_SL > 0.5

Fatal errors: PWR\_LAW\_EXPNT\_SL < 0.0

PWR\_LAW\_EXPNT\_SL > 10.0

## **REF HGT**

Floating point value representing the height (m) above terrain to which the surface wind observations are adjusted using the surface power law (PWR\_LAW\_EXPNT\_SL). Adjustments of the measurements to a common height, usually the typical height of the surface instruments, permits the extrapolation of winds in the horizontal direction.

Warning errors: REF\_HGT ≤ 2.0

REF\_HGT ≥ 100.0

Fatal errors: REF\_HGT ≤ 0.0

REF\_HGT ≥ 500.0

## **RWMDATE**

7-character string specifying the valid date (ddmmmyy) of the Relocatable Window Model gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

## **RWMFCST**

Integer specifying the forecast period (hh) of the Relocatable Window Model gridded metdata to be used by MEDIC. For example, a value of 00 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the Relocatable Window Model are 00, 06, 24, and 36. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

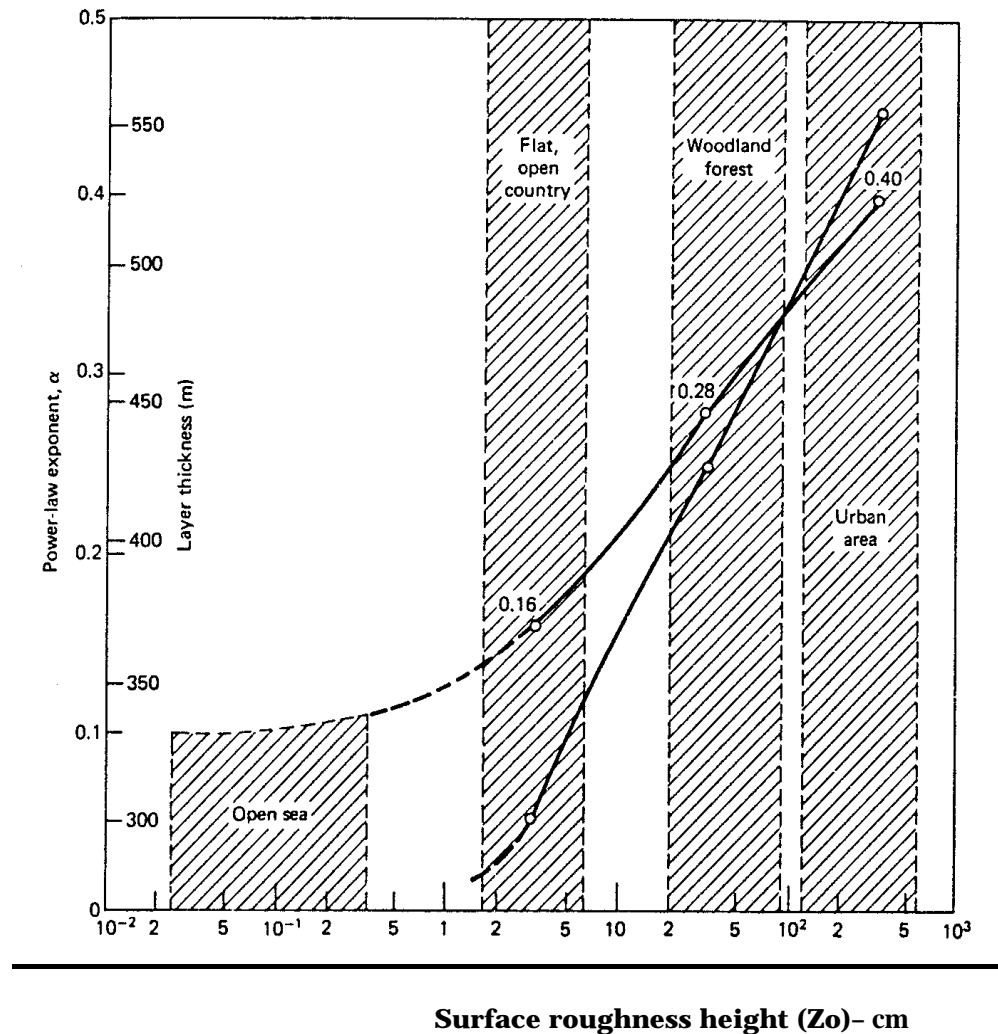


Figure III.B.1 Modeling the planetary boundary layer (plot from Davenport 1965)

## **RWMTIME**

8-character string specifying the valid time (hh:mm:ss) of the Relocatable Window Model gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**SHEDATE** (MEDIC/HMEDIC variable)

7-character string specifying the valid date (ddmmmyy) of the Southern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**SHEFCST** (MEDIC/HMEDIC variable)

Integer specifying the forecast period (hh) of the Southern Hemisphere gridded metdata to be used by MEDIC. For example, a value of 00 refers to a zero-hour forecast or initialization; a value of 24 refers to a 24-hour forecast. Current possible values for the hemispheric gridded data are 00, 12, 24, 36, and 48. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**SHETIME** (MEDIC/HMEDIC variable)

8-character string specifying the valid time (hh:mm:ss) of the Southern Hemisphere gridded metdata to be used by MEDIC. The gridded metdata is transformed into upper air observations for the area covered by the MEDIC grid.

**SINGLE PROFILE WINDS**

8-character string ('Y', met station id, 'N') controlling whether or not a single wind profile is to be used. 'Y' = single wind profile to be specified via ALL\_WINDS\_SPD, ALL\_WINDS\_DIR, and ALL\_WINDS\_HGT; met station id = single wind profile for the specified metdata station id from the Problem Metdata File; or 'N' = spatially-varying winds (specified via Problem Metdata Files, OBSERV.MET file, or gridded data). Interpolation is done linearly, with rotation sense determined by the MEDIC parameter MAX\_VEER\_PARAM\_VERT, and following existing direction interpolation rules. SINGLE\_PROFILE\_WINDS must be set to 'Y' in order to use ALL\_WINDS\_SPD, ALL\_WINDS\_DIR, and ALL\_WINDS\_HGT.

(Default: SINGLE\_PROFILE\_WINDS = 'N')

**SL\_HGT**

Floating point value representing the height (m) of the surface layer above terrain. The surface boundary layer is defined as that part of the atmosphere boundary layer where momentum, heat, and energy fluxes are constant. While the height of this layer is typically about 10% of the boundary layer height (BL\_HGT), it is more useful to consider SL\_HGT as the level at which the wind direction begins to turn with height.

Warning errors: SL\_HGT  $\leq$  2.0  
SL\_HGT  $\geq$  100.0  
Fatal errors: SL\_HGT  $\leq$  0.0  
SL\_HGT  $\geq$  500.0

## **USE UPR AT SFC**

Flag ('Y' or 'N') which controls whether the lowest level of upper air observations from all sources which are also within the surface layer is used to construct the reference level (surface) winds.

(Default : USE\_UPR\_AT\_SFC = 'N')

## **SAMPLE MEDIC.NML INPUT FILE FOR HEMISPHERIC CASE**

<b>\$medic_control</b>			beginning of medic control namelist
medic_vlst	= 'new'	←	flag controlling creation of MEDIC.VLST file
print_windfields	= 'Y'	←	flag controlling output of winds
\$end		←	end of medic control namelist
<b>\$medic_params</b>			beginning of medic params namelist
met_start_date	= '01APR85'	←	date(ddMMMyy) of medvel file to be created
met_start_time	= '0600'	←	time (hhmmss) of medvel file to be created
avgttime	= '0100'	←	averaging time for medvel file to be created
nhedate	= '01APR85'	←	date (ddMMMyy) of AFGWC met dataset to be used
nhfcst	= 00	←	analysis (00hr) AFGWC data to be used
nhetime	= '1200'	←	time ('hhmmss') of AFGWC met dataset to be used
\$end		←	end of medic params namelist

## SAMPLE MEDIC INPUT FILE USING OBSERV.MET DATA

<b>\$medic_control</b>		beginning of control namelist
medic_vlst	= 'new'	← flag controlling creation of MEDIC.VLST file
print_windfields	= 'Y'	← prints wind components
sfc_rough_hgt	= 0.1	← surface roughness height
Send		← end of control namelist
 <b>\$medic_params</b>		beginning of params namelist
metdataset	= 'SET1'	← pointer id for met dataset in OBSERV.MET
met_start_date	= '01APR85'	← date(ddMMMy) of medvel file to be created
met_start_time	= '0600'	← time (hhmmss) of medvel file
avgtime	= '0100'	← averaging time for medvel file
sl_hgt	= 25.0	← height of the surface layer above the terrain
bl_hgt	= 1400.0	← height of the boundary layer above the terrain
pwr_low_expnt_sl	= 0.30	← exponent for surface layer power law
extrap_expnt_bl	= 0.30	← extrapolation exponent for boundary layer
profile_extrap	= 'N'	← method of extrapolation
ref_hgt	= 10.0	← hgt above terrain to which sfc winds are adjusted
Send		← end of params namelist

## SAMPLE MEDIC.NML INPUT FILE USING PROBLEM MET DATA

<b>\$medic_control</b>		beginning of medic control namelist
medic_vlst	= 'new',	← flag controlling creation of MEDIC.VLST file
print_windfields	= 'Y'	← prints wind components
sfc_rough_hgt	= 0.1	← surface roughness height
Send		← end of medic control namelist
 <b>\$medic_params</b>		beginning of medic params namelist
probmet_dst	= '850401060'	← specification of problem met data file
met_start_date	= '01APR85'	← date(ddMMMy) of medvel file to be created
met_start_time	= '0600'	← time (hhmmss) of medvel file to be created
avgtime	= '0100'	← averaging time for medvel file to be created
sl_hgt	= 25.0	← height of the surface layer above the terrain
bl_hgt	= 1400.0	← height of the boundary layer above the terrain
pwr_low_expnt_sl	= 0.30	← exponent for surface layer power law
extrap_expnt_bl	= 0.30	← extrapolation exponent for boundary layer
profile_extrap	= 'N'	← method of extrapolation
ref_hgt	= 10.0	← hgt above terrain to which sfc winds are adjusted
Send		← end of medic params

## SAMPLE MEDIC INPUT FILE USING SINGLE\_PROFILE\_WINDS = 'Y'

<b>\$medic_control</b>		beginning of control namelist
medic_vlst	= 'new'	← flag controlling creation of MEDIC.VLST file
print_windfields	= 'Y'	← prints wind components
sfc_rough_hgt	= 0.1	← surface roughness height
stnlocpsl	= 'N'	← no problem station library info required
Send		← end of control namelist



<b>\$medic_params</b>			beginning of params namelist
metdataset	=	'SET1'	← pointer id for met dataset in OBSERV.MET
met_start_date	=	'01APR85'	← date(ddMMMy) of medvel file to be created
met_start_time	=	'0600'	← time (hhmmss) of medvel file
avgtime	=	'0100'	← averaging time for medvel file
single_profule_winds	=	'Y'	← using single wind profile specified by ALL_WINDS...
all_winds_hgt	=	10. 100. 150. 300. 500	← array of heights AGL (m)
all_winds_dir	=	160. 180. 182. 185. 190	← array of wind directions (deg)
all_winds_dpc	=	1.5 2.0 2.3 3.0 4.5	← array of wind speeds (m/s)
sl_hgt	=	25.0	← height of the surface layer above the terrain
bl_hgt	=	1400.0	← height of the boundary layer above the terrain
pwr_low_expnt_sl	=	0.30	← exponent for surface layer power law
extrap_expnt_bl	=	0.30	← extrapolation exponent for boundary layer
profile_extrap	=	'N'	← method of extrapolation
ref_hgt	=	10.0	← hgt above terrain to which sfc winds are adjusted
\$end			← end of params namelist

### SAMPLE MEDIC.NML INPUT FILE USING SINGLE\_PROFILE\_WINDS = 'metstnid'

<b>\$medic_control</b>			beginning of medic control namelist
medic_vlst	=	'new',	← flag controlling creation of MEDIC.VLST file
print_windfields	=	'Y'	← prints wind components
sfc_rough_hgt	=	0.1	← surface roughness height
\$end			← end of medic control namelist
<b>\$medic_params</b>			beginning of medic params namelist
probmest_dst	=	'850401060'	← specification of problem met data file
met_start_date	=	'01APR85'	← date(ddMMMy) of medvel file to be created
met_start_time	=	'0600'	← time (hhmmss) of medvel file to be created
avgtime	=	'0100'	← averaging time for medvel file to be created
single_profile_winds	=	'ARACTES1'	← use single wind profile for station ARACTES1 from specified probmet_dst file
sl_hgt	=	25.0	← height of the surface layer above the terrain
bl_hgt	=	1400.0	← height of the boundary layer above the terrain
pwr_low_expnt_sl	=	0.30	← exponent for surface layer power law
extrap_expnt_bl	=	0.30	← extrapolation exponent for boundary layer
profile_extrap	=	'N'	← method of extrapolation
ref_hgt	=	10.0	← hgt above terrain to which sfc winds are adjusted
\$end			← end of medic params

### III.B.5. STNLOC.MET

The STNLOC.MET file is the input file containing the met station information for the MEDIC code when the Problem Station Library file is not used. It specifies meteorological measurement stations, both surface and upper air, by name, location, and (for the surface stations) measurement height. Each meteorological measurement station referenced in either the Problem Metadata Files or the OBSERV.MET file must have been assigned a location in the STNLOC.MET file if there is no Problem Station Library. While the STNLOC.MET file normally is left unchanged for a well-characterized site, it can be changed at any time between MEDIC runs to provide locations for additional meteorological stations that have become available during an accident response.

This file is a Fortran list-directed input file (rather than a Fortran namelist file). Preceding station location records is a keyword identifying which type of station follows: SFC or UPR, corresponding to surface or upper air meteorological data stations. The records following the keyword have no identifying names, but order is important. The data to specify a surface location is the name (8 characters), x coordinate, y coordinate, and height. An upper air location is similarly specified, except a height is not specified.

The following variables must be assigned a value in the STNLOC.MET file.

#### **Surface Location Record**

There is one record per surface station (maximum of 200), and the record consists of four items.

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. The array for this value in the code is NAMESFC.
2. The Model x coordinate of the surface station location in kilometers. The array for this value in the code is SFCSTX.
3. The Model y coordinate of the surface station location in kilometers. The array for this value in the code is SFCSTY.
4. The height of the instrument above ground level in meters. The array for this value in the code is SFCHGT.

Fatal errors:

- SFCHGT  $\geq$  100.0
- SFCSTX  $<$  -45000.0
- SFCSTY  $<$  -45000.0
- SFCHGT  $\leq$  0.0
- SFCSTX  $\geq$  45000.0
- SFCSTY  $\geq$  45000.0
- SFCHGT  $\geq$  10000.0

### **Upper Air Location Record**

There is one record per upper air station (maximum of 100), and the record consists of three items:

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. The array for this value in the code is NAMEUPR
2. The Model x coordinate of the upper air station location in kilometers. The array for this value in the code is UPRSTX.
3. The Model y coordinate of the upper air station location in kilometers. The array for this value in the code is UPRSTY.

Fatal errors:      UPRSTX < -45000.0

                     UPRSTY < -45000.0

                     UPRSTX ≥ 45000.0

                     UPRSTY ≥ 45000.0

No entry is required for the number of either surface or upper air records; a count of valid stations of each type is done in the input processing.

### **SAMPLE STNLOC.MET FILE**

UPR				
	'745040'	544.196	4150.254	
	'OAK'	569.007	4178.162	
SFC				
	'HWD'	577.921	4167.145	6.1
	'LVK'	604.317	4172.984	6.1
	'OAK'	569.022	4176.308	6.1
	'NGZ'	560.170	4181.787	6.1
	'SCK'	653.864	4195.958	6.1
	'CCR'	583.429	4204.182	6.1
	'ARACTES1'	613.496	4171.685	10.0
	'LIVELLNL'	614.661	4172.278	10.0

### **III.B.6. OBSERV.MET**

OBSERV.MET is the input file used when meteorological data is not obtained from the Problem Metdata Files. The file is constructed to be read as a Fortran list-directed file, as was STNLOC.MET. An OBSERV.MET file can contain many individual meteorological data sets, called a metdataset. Each such set of data is marked by the occurrence of the keyword METDATASET, followed on the same line by the name (up to 40 characters), in apostrophes, by which it is to be referred in this MEDIC run. Following the METDATASET identifier record is either of the keywords UPR or SFC, which denote that the meteorological data to follow is either upper air or surface data respectively. Following the type-identifier record are individual data records. When the keywords UPR or SFC are again encountered, then input processing toggles to that type. When the METDATASET keyword is encountered, a new set of data is deemed to begin, terminating the previous input one. The format of both types of individual data records is now given.

#### **Surface Met Record**

There is one record for each surface observation, and this record consists of three items.

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. This station identifier name must match exactly with one previously input as a surface station via the STNLOC.MET file or from the Problem Station Library.
2. The wind direction (degrees) is entered. If no wind direction data (i.e. this record is only for a wind speed entry), enter -1.
3. The wind speed (meters/sec) is entered. If no wind speed data (i.e., this record is only for a wind direction entry), enter -1. A zero wind speed is assigned a -1. direction.

#### **Upper Air Met Record**

There is one record for each upper air level (maximum of 35 levels per upper air station), and the record consists of four items.

1. The station identifier, which is a character string of up to eight (8) characters delimited by apostrophes. This station identifier name must match exactly with one previously input as an upper air station via the STNLOC.MET file or from the Problem Station Library. This must be specified for the first upper air level, while subsequent levels/records for the same station may simply contain a comma instead of the station identifier. (See example on next page)
2. The observation height, in meters (required).
3. The wind direction (degrees) is entered. If no wind direction data (i.e. this record is only for a wind speed entry), enter -1.
4. The wind speed (meters/sec) is entered. If no wind speed data (i.e., this record is only for a wind direction entry), enter -1. A zero wind speed is assigned a -1. direction.
5. Note: sigma theta data is entered only via the SIGMA\_THETA\_DATA namelist for ADPIC.

# **SAMPLE OBSERV.MET FILE**

METDATASET	'SET1'			
UPR				
	'OAK'	10.0	270.0	1.0
		, 500.0	270.0	3.0
		, 900.0	270.0	3.0
		, 1400.0	270.0	3.0
SFC				
	'CCR'		270.	2.5
	'HWD'		325.	1.0
	'LVK'		270.	1.0
	'SCK'		350.	1.0
	'LIVELLNL'		250.	1.0

### **III.B.7. CG-MATHEW.NML**

CG-MATHEW.NML is the input file that controls a CG-MATHEW run. A basic CG-MATHEW.NML file is a list of the CG-MATHEW stabilities (or SIGMA\_VERT values) and the date/times when they take effect. The CG-MATHEW.NML file may also contain a list of the MEDVEL\*.BIF files to be processed although these are more typically specified in a MEDIC.VLST file.

The following is a list of the CG-MATHEW.NML variables grouped alphabetically within namelist blocks:

#### **III.B.7.a. CG-MATHEW\_CONTROL namelist**

##### **HORZ VECTOR SCALE**

Floating point value which scales the size of the vectors on the vector plots in the  $x$  direction.

(Default: HORZ\_VECTOR\_SCALE = 0.5).

Warning errors:  $\text{HORZ\_VECTOR\_SCALE} \geq 25.0$

Fatal errors:  $\text{HORZ\_VECTOR\_SCALE} \leq 0.0$

##### **IPROFILE**

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard grid) of the  $i$  indices of the sample profiles of the adjusted winds to be plotted. The first value in IPROFILE is matched with the first value in JPROFILE to give the horizontal grid indices of the first profile. Succeeding indices from each list are matched in a similar fashion. The default for IPROFILE with a standard 51 by 51 by 15 CG-MATHEW grid is: 14 26 38 14 26 38 14 26 38. This gives the normal nine sample profiles. Any profiles specified by IPROFILE and JPROFILE (along with the defaults) are superceded by any locations specified by XPROFILE and YPROFILE.

##### **ISLICE**

Array (max size = IMAXMAT = 51) of integers (1 to 51 for standard grid) of  $i$ -indices at which vector plots of  $v - w$  component winds are to be produced. The default value is no slices. Any slices specified by ISLICE are superceded by any locations specified by XSLICE.

##### **ITER MESSAGE**

Integer value specifying the interval number of iterations at which status messages are sent to the terminal. The message gives the number of iterations and the number of “cells” that must still converge.

(Default: ITER\_MESSAGE = 50).

Warning errors: None

Fatal errors: ITER\_MESSAGE < 0

### **JPROFILE**

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard grid) of the  $j$  indices of the sample profiles of the adjusted winds to be plotted. The first value in JPROFILE is matched with the first value in IPROFILE to give the horizontal grid indices of the first profile. Succeeding indices from each list are matched in a similar fashion. The default for JPROFILE with a standard 51 by 51 by 15 CG-MATHEW grid is: 14 14 14 26 26 26 38 38 38. This gives the normal nine sample profiles. Any profiles specified by IPROFILE and JPROFILE (along with the defaults) are superceded by any locations specified by XPROFILE and YPROFILE.

### **JSLICE**

Array (max size = JMAXMAT = 51) of integers (1 to 51 for standard grid) of  $j$  indices at which vector plots of  $u - w$  component winds are to be produced. The default value is no slices. Any slices specified by JSLICE are superceded by any locations specified by YSLICE.

### **OUTPUT DESTINATION DIR**

10-character string ('WINDS', 'DISPERSION', 'VLST', or 'GRID') which directs all output from CG-MATHEW (except the CG-MATHEW.LOG file) to a specified directory.

(Default: OUTPUT\_DESTINATION\_DIR = 'WINDS')

### **PLOT LEVELS ABOVE TOPO**

Array (max size = KMAXMAT = 15) of integers (1 to 15) representing the  $k$  levels above terrain at which horizontal vector plots are to be produced.

(Default: PLOT\_LEVELS\_ABOVE\_TOPO = 2 -> produce vector plot at 1 \*DELZ above terrain)

### **PLOT GRID LEVELS**

Array (max size = KMAXMAT = 15) of integers (1 to 15) representing the  $k$  levels above grid bottom (thus some areas may contain vectors of zero length corresponding to grid points underground) at which horizontal vector plots are to be produced. The default is no plots.

### **PRINT ADJUSTWIND ADPIC**

Flag ('Y', 'N') specifying whether or not ADPIC winds are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_ADJUSTWIND\_ADPIC = 'N' -> don't print ADPIC winds in CG-MATHEW.LOG)

### **PRINT ADJUSTWIND DIVERG**

Flag ('Y', 'N') specifying whether or not the divergence of the adjusted grid point ADPIC winds are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_ADJUSTWIND\_DIVERG = 'N' -> don't print the divergence of the adjusted grid point ADPIC winds in the CG-MATHEW.LOG file)

#### **PRINT\_ADJUSTWIND\_FACE**

Flag ('Y', 'N') specifying whether or not the adjusted face-centered winds are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_ADJUSTWIND\_FACE = 'N' -> don't print these winds in the CG-MATHEW.LOG)

#### **PRINT\_ADJUSTWIND\_GRID**

Flag ('Y', 'N') specifying whether or not the grid point adjusted winds are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_ADJUSTWIND\_GRID = 'N' -> don't print these winds in the CG-MATHEW.LOG)

#### **PRINT\_CONVERGENCE**

Flag ('Y' or 'N') specifying whether or not the convergence status is to be displayed to the screen during iteration.

(Default: PRINT\_CONVERGENCE = 'Y')

#### **PRINT\_ICCG\_CHECK**

Flag ('Y' or 'N') which controls the printing of the divergence checks on the conjugate-gradient solution.

(Default = PRINT\_ICCG\_CHECK = 'N')

#### **PRINT\_INITIALWIND\_DIVERG**

Flag ('Y', 'N') specifying whether or not the divergence of the extrapolated winds is to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_INITIALWIND\_DIVERG = 'N' -> don't print the divergence in the CG-MATHEW.LOG)

#### **PRINT\_INITIALWIND\_FACE**

Flag ('Y', 'N') specifying whether or not the face-centered extrapolated winds are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_INITIALWIND\_FACE = 'N' -> don't print these winds in the CG-MATHEW.LOG)

#### **PRINT\_INITIALWIND\_GRID**

Flag ('Y', 'N') specifying whether or not the grid point extrapolated winds are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_INITIALWIND\_GRID = 'N' -> don't print these winds in the CG-MATHEW.LOG)



### **PRINT LAMBDAS**

Flag ('Y', 'N') specifying whether or not the Lagrangian multipliers are to be printed in the CG-MATHEW.LOG file.

(Default: PRINT\_LAMBDAS = 'N' -> don't print the multipliers in the CG-MATHEW.LOG)

### **PRINT SIDES**

Flag ('Y' or 'N') which controls the printing of the three-dimensional arrays identifying the cell side classifications.

(Default: PRINT\_SIDES = 'N')

### **VECTOR PLOT FREQ**

Integer value controlling the density of vectors on the vector plots. A value of one (1) indicates that a vector is plotted at every horizontal grid point. A value of two indicates that a vector is to be plotted every other grid point, etc.

(Default: VECTOR\_PLOT\_FREQ = 2)

Warning errors: None

Fatal errors: VECTOR\_PLOT\_FREQ < 0

### **VERT VECTOR SCALE**

Floating point value which scales the size of the vectors on the vector plots in the *z* direction.

(Default: VERT\_VECTOR\_SCALE = 25.0)

Warning errors: VERT\_VECTOR\_SCALE  $\geq$  200.0

Fatal errors: VERT\_VECTOR\_SCALE  $\leq$  0.0

### **VLST OPTION**

6-character string ('NONE', 'NEW', 'MODIFY') specifying how CG-MATHEW is to handle output of a VLST file. If it is set to 'NONE ' then CG-MATHEW does not produce a CG-MATHEW.VLST file, this is the default. If it is 'NEW ' then CG-MATHEW produces a CG-MATHEW.VLST file that supercedes any preceeding versions. If it is set to 'MODIFY' then CG-MATHEW reads the CG-MATHEW.VLST file currently available (if it exists) and the overwrites MATVEL filenames that match those being processed in the current run except for the sequence number and appends new filenames to the end of the CG-MATHEW.VLST records.

### **XPROFILE**

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard grid) of the *x* coordinates of the sample profiles of the adjusted winds to be plotted. The first value in XPROFILE is matched with the first value in YPROFILE to give a horizontal location for the first profile. The profile location

plotted is the  $i,j$  grid location nearest to the specified  $x,y$  coordinates. Succeeding coordinates from each list are matched in a similar fashion. Values for XPROFILE, YPROFILE supercede any user settings for IPROFILE and JPROFILE as well as the nine default profiles (see IPROFILE and JPROFILE for the defaults). Each XPROFILE value must lie within the CG-MATHEW grid.

### **XSLICE**

Array (max size = IMAXMAT = 51) of floating point values representing the  $x$  coordinates of the vertical planes of  $v-w$  winds vectors of the adjusted winds to be plotted. The slice location plotted is the  $i$  grid location nearest to the specified  $x$  coordinate. Values for XSLICE supercede any user settings for ISLICE. The default is no slice plots. Each XSLICE value must lie within the CG-MATHEW grid.

### **YPROFILE**

Array (max size = MAXPROF = 10) of integers (1 to 51 for standard grid) of the  $y$  coordinates of the sample profiles of the adjusted winds to be plotted. The first value in YPROFILE is matched with the first value in XPROFILE to give a horizontal location for the first profile. The profile location plotted is the  $i,j$  grid location nearest to the specified  $x,y$  coordinates. Succeeding coordinates from each list are matched in a similar fashion. Values for XPROFILE, YPROFILE supercede any user settings for IPROFILE and JPROFILE as well as the nine default profiles (see IPROFILE and JPROFILE for the defaults). Each YPROFILE value must lie within the CG-MATHEW grid.

### **YSLICE**

Array (max size = JMAXMAT = 51) of floating point values representing the  $y$  coordinates of the vertical planes of  $u-w$  winds vectors of the adjusted winds to be plotted. The slice location plotted is the  $j$  grid location nearest to the specified  $y$  coordinate. Values for YSLICE supercede any user settings for JSLICE. The default is no slice plots. Each YSLICE value must lie within the CG-MATHEW grid.



### **III.B.7.b. CG-MATHEW\_PARAMS namelist**

#### **CONV\_TOLERANCE**

Floating point value specifying the required convergence tolerance limit for the conjugate-gradient solver.

(Default: CONV\_TOLERANCE = 0.01)

#### **ELLIPTIC\_EQN\_TEST**

Flag ('Y' or 'N') which controls whether the code sets up and runs an elliptic equation test based on sinusoidal fields to test the implementation of boundary conditions and the conjugate-gradient solver.

(Default: ELLIPTIC\_EQN\_TEST = 'N')

#### **FIX\_LATERAL\_INFLOW**

Flag ('Y' or 'N') which controls whether the inflow is fixed on all lateral faces.

(Default: FIX\_LATERAL\_INFLOW = 'N')

#### **FIX\_XMAX\_INFLOW**

Flag ('Y' or 'N') which controls whether the inflow is fixed on the maximum index x face. Overridden if FIX\_LATERAL\_INFLOW is set.

(Default: FIX\_XMAX\_INFLOW = 'N')

#### **FIX\_XMIN\_INFLOW**

Flag ('Y' or 'N') which controls whether the inflow is fixed on the minimum index x face. Overridden if FIX\_LATERAL\_INFLOW is set.

(Default: FIX\_XMIN\_INFLOW = 'N')

#### **FIX\_YMAX\_INFLOW**

Flag ('Y' or 'N') which controls whether the inflow is fixed on the maximum index y face. Overridden if FIX\_LATERAL\_INFLOW is set.

(Default: FIX\_YMAX\_INFLOW = 'N')

#### **FIX\_YMIN\_INFLOW**

Flag ('Y' or 'N') which controls whether the inflow is fixed on the minimum index y face. Overridden if FIX\_LATERAL\_INFLOW is set.

(Default: FIX\_YMIN\_INFLOW = 'N')

### **ITER HALT**

Integer value specifying the maximum allowed number of iterations for the conjugate gradient solver.

(Default: ITER\_HALT = 1500)

### **LATERAL BOUNDARY**

15-character string('ADJUST\_NORMAL', 'FIXED\_NORMAL', or 'MIXED') specifying the boundary condition on all lateral faces.

(Default: LATERAL\_BOUNDARY = 'ADJUST\_NORMAL')

### **MEDVEL NAMES**

Array (max size = MAX\_VEL = 10000) of 40-character strings that specify the MEDVEL\*\*\*.BIF files to be processed by the current CG-MATHEW run. Each string is a fully qualified, and thus unique, filename. The list of files entered via MEDVEL\_NAMES supercedes the list of files specified in the MEDIC.VLST file, an alternative way of indicating those MEDVEL\*\*\*.BIF files to be processed.

### **MIXED LATERAL ADJUST**

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on all lateral faces.

(Default: MIXED\_LATERAL\_ADJUST = 1.0)

### **MIXED TOP ADJUST**

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the top face.

(Default: MIXED\_TOP\_ADJUST = 1.0)

### **MIXED XMAX ADJUST**

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index x face. Overridden if MIXED\_LATERAL\_ADJUST is set.

(Default: MIXED\_XMAX\_ADJUST = 1.0)

### **MIXED XMIN ADJUST**

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index x face. Overridden if MIXED\_LATERAL\_ADJUST is set.

(Default: MIXED\_XMIN\_ADJUST = 1.0)

### **MIXED YMAX ADJUST**

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index y face. Overridden if MIXED\_LATERAL\_ADJUST is set.

(Default: MIXED\_YMAX\_ADJUST = 1.0)

### **MIXED YMIN ADJUST**

Floating point value in inverse meters specifying the relative amount of normal to tangential adjustment for MIXED boundary conditions on the maximum index y face. Overridden if MIXED\_LATERAL\_ADJUST is set.

(Default: MIXED\_YMIN\_ADJUST = 1.0)

### **PARAMS DATE**

Array (max size = MAX\_TIMES = 10000) of 7-character strings specifying the dates (ddmmmyy) for the time-varying inputs, STAB\_CG-MATHEW and SIGMA\_VERT, i.e., it determines the date component of the date/time when one or the other of these variables (but not both) change values.

### **PARAMS TIME**

Array (max size = MAX\_TIMES = 10000) of 8-character strings specifying the times (hh:mm:ss) for the time-varying inputs, STAB\_CG-MATHEW and SIGMA\_VERT, i.e., it determines the time component of the date/time when one or the other of these variables (but not both) change values.

### **SIGMA HORZ**

Floating point value that, together with SIGMA\_VERT, controls how much mass-adjustment takes place in the horizontal as opposed to the vertical direction. Since the ratio of SIGMA\_VERT to SIGMA\_HORZ controls CG-MATHEW's behavior, SIGMA\_HORZ is normally set to 1.0 and only SIGMA\_VERT is changed. The default value is 1.0.

Warning errors: SIGMA\_HORZ > 5.0

Fatal errors: SIGMA\_HORZ ≤ 0.0

SIGMA\_HORZ ≥ 100.0

### **SIGMA VERT**

Array(max size = MAX\_TIMES=10,000) of floating point values that, together with SIGMA\_HORZ, control how much mass-adjustment takes place in the vertical as opposed to the horizontal direction. The choice of SIGMA\_VERT is a function of the grid cell geometry and the atmospheric stability and can be associated, all be it crudely, with Pasquill categories (see CG-MATHEW input variable STAB\_CG-MATHEW below). The assessor may use SIGMA\_VERT to gain finer control over the CG-MATHEW adjustment process than is provided via STAB\_CG-MATHEW. SIGMA\_VERT is a time-varying input like STAB\_CG-MATHEW, thus any change in its value must be associated with corresponding entries in the PARAMS\_DATE and PARAMS\_TIME arrays.

Warning errors: SIGMA\_VERT > 2.0  
 Fatal errors: SIGMA\_VERT ≤ 0.0  
 SIGMA\_VERT > 100.0

### **STAB CG-MATHEW**

Array (max size = MAX\_TIMES = 10000) of integers (1-6) representing the Pasquill-Gifford stability classes that, when associated with the PARAMS\_DATE and PARAMS\_TIME arrays, allows CG-MATHEW to determine the appropriate stability for any given data set it is processing. Thus, STAB\_CG-MATHEW can be considered a time-varying input analogous to those in the ADPIC.NML file. These times need only be specified when STAB\_CG-MATHEW changes, not necessarily for every data set. STAB\_CG-MATHEW is used to control the relative degree of adjustment by assigning values to SIGMA\_VERT according to the table below. Thus, SIGMA\_VERT is a more direct way of controlling CG-MATHEW's behavior and is also a time-varying input array. The use of SIGMA\_VERT directly supercedes the use of STAB\_CG-MATHEW.

(Default: STAB\_CG-MATHEW = 4 -> neutral stability)

STAB_CLASS	SIGMA_VERT
1	$3.3 \times (2 \times \text{DELZ} / \text{DELX})$
2	$2.4 \times (2 \times \text{DELZ} / \text{DELX})$
3	$1.6 \times (2 \times \text{DELZ} / \text{DELX})$
4	$1.0 \times (2 \times \text{DELZ} / \text{DELX})$
5	$0.7 \times (2 \times \text{DELZ} / \text{DELX})$
6	$0.33 \times (2 \times \text{DELZ} / \text{DELX})$

Fatal errors: STAB\_CG-MATHEW < 1  
 STAB\_CG-MATHEW > 6

A sample CG-MATHEW namelist file is shown below:

### **TOP BOUNDARY**

15-character string ('ADJUST\_NORMAL', 'FIXED\_NORMAL', or 'MIXED') specifying the boundary condition on the top face (Default: TOP\_BOUNDARY = 'ADJUST\_NORMAL').

SCG-MATHEW_params		
print_adjustwind_diverg	= 'Y'	← print adjusted divergence
print_initialwind_diverg	= 'Y'	← print initial divergence
vector_plot_freq	= 1	← plot all wind vectors
Send		
SCG-MATHEW_params		
MEDVEL_NAMES	=	← datasets to be processed.
'MEDVEL_01APR85_0000_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0100_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0200_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0300_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0400_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0500_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0600_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0700_00_000_0100_012.BIF'		
'MEDVEL_01APR85_0800_00_000_0100_012.BIF'		
PARAMS_DATE = '01APR85' '01APR85'		
		← date of new stabilities
PARAMS_TIME = '0000' '0500'		← time of new stabilities
STAB_CG-MATHEW = 3 5		← new stabilities
Send		

### **XMAX BOUNDARY**

15-character string ('ADJUST\_NORMAL', 'FIXED\_NORMAL', or 'MIXED') specifying the boundary condition on the maximum index x face. Overridden if LATERAL\_BOUNDARY is set.

(Default: XMAX\_BOUNDARY = 'ADJUST\_NORMAL')

### **XMIN BOUNDARY**

15-character string ('ADJUST\_NORMAL', 'FIXED\_NORMAL', or 'MIXED') specifying the boundary condition on the minimum index x face. Overridden if LATERAL\_BOUNDARY is set.

(Default: XMIN\_BOUNDARY = 'ADJUST\_NORMAL')

### **YMAX BOUNDARY**

15-character string ('ADJUST\_NORMAL', 'FIXED\_NORMAL', or 'MIXED') specifying the boundary condition on the maximum index y face. Overridden if LATERAL\_BOUNDARY is set.

(Default: YMAX\_BOUNDARY = 'ADJUST\_NORMAL')

### **YMIN BOUNDARY**

15-character string ('ADJUST\_NORMAL', 'FIXED\_NORMAL', or 'MIXED') specifying the boundary condition on the minimum index y face. Overridden if LATERAL\_BOUNDARY is set.

(Default: YMIN\_BOUNDARY = 'ADJUST\_NORMAL')



### **III.B.8. MEDIC.VLST**

CG-MATHEW can optionally read a list of fully qualified, and therefore unique, MEDVEL\*\*\*.BIF filenames from MEDIC.VLST. This file is produced by MEDIC and contains the list of MEDVEL\*\*\*.BIF files created by a given MEDIC run. This is an ASCII file that can be edited by an assessor and provides a mechanism to handle multiple MEDVEL groups in a single directory by allowing the user to maintain a series of MEDIC.VLST files (under different names, with the desired file being copied to MEDIC.VLST as needed).



### **III.B.9. ADPIC.NML**

ADPIC.NML is an input file to the ADPIC code which contains the source and editing parameter data for an ADPIC problem. Before ADPIC is run, either initially or as a restart, an ADPIC.NML file must be prepared.

Descriptions of all of the individual ADPIC.NML parameters follows, by the namelists in which they are to be entered:

#### **III.B.9.a ADPIC\_CONTROL namelist**

##### **DIFF METHOD**

5-character string ('GRADI' or 'MC1') specifying the type of diffusion calculation method to be used. 'GRADI' refers to gradient or K-theory diffusion; 'MC1' refers to a Monte-Carlo or Markov-chain diffusion method using the Markov equation for turbulent velocity based upon the Langevin equation. The ADPIC parameter TURB\_PARAM\_TYPE controls the diffusion coefficients to be used. Note: The Monte Carlo diffusion method (option 'MC1') is still under evaluation.

##### **INITIAL TIME STEP**

Floating point value specifying the value for the initial time step (seconds) for ADPIC; all succeeding time steps will be calculated by the code. ADPIC calculates the succeeding time steps such that a particle should not move more than one cell in that time step. INITIAL\_TIME\_STEP is also used to generate a warning error ("Time step has decreased to x.xxxExx") whenever code-calculated time steps are less than INITIAL\_TIME\_STEP/10.

##### **NESTS**

Integer (0-4) controlling the number of sampling nests to be used. fewer nests require less buffer zone between the source and the edge of the grid. The default center of the nests will be the least squares center of all the source locations. This center can be overridden with the ADPIC\_CONTROL namelist parameters X\_NEST\_LOC and Y\_NEST\_LOC.

(Default: NESTS = 4)

##### **PARTPOS**

A 1-character string ('N', 'P', 'V') indicating in which format to generate PARTPOS files at the time interval specified by PLOT\_INTERVAL:

'N' = no PARTPOS file is to be generated

'P' = PERSPEC format (3 particle position arrays)

'V' = video format (3 particle position arrays, plus source and identifier arrays)

(Default: PARTPOS = 'N' → no PARTPOS files are to be generated)

### **PLOT INTERVAL**

A 13-character time interval string (ddd::hhmmss) specifying the time interval at which particle plots are written to the graphics file.

(Default: PLOT\_INTERVAL = '000::01:00:00' → 1-hour plot interval)

### **PRINT ADVECT VEL**

Flag ('Y' or 'N') which controls the printing of advection velocities in echo file as they are read from the MATVEL file.

(Default: PRINT\_ADVECT\_VEL = 'N' → don't print advection velocities)

### **PRINT CELL CONC**

Flag ('Y' or 'N') which controls the printing of cell concentrations in echo file at each SAMPLING\_INTERVAL interval.

(Default: PRINT\_CELL\_CONC = 'N' → print cell concentrations)

### **PRINT FRCTN VEL**

Flag ('Y' or 'N') which controls the printing of friction velocities in the echo file at each STATUS\_INTERVAL interval.

(Default: PRINT\_FRCTN\_VEL = 'N' → don't print friction velocities)

### **PRINT MC DIAG**

Flag ('Y' or 'N') which controls the printing of extensive Monte Carlo diagnostics.

Note: This parameter used ONLY for DIFF\_METHOD = 'MC1'.

(Default: PRINT\_MC\_DIAG = 'N')

### **PRINT PART COORD**

Flag ('Y' or 'N') which controls the printing of particle coordinates in echo file at each SAMPLING\_INTERVAL interval.

(Default: PRINT\_PART\_COORD = 'N' → don't print particle coordinates)

### **PRINT SAMPLING BINS**

Flag ('Y' or 'N') which controls the printing of sampling bin concentrations to the echo file at each SAMPLING\_INTERVAL interval.

(Default: PRINT\_SAMPLING\_BINS = 'N' → don't print sampling bin concentrations)

### **PRINT TURB PARAM GRADI**

Flag ('Y' or 'N') which controls the printing (to the ADPIC log file) of horizontal and vertical diffusion coefficients (Kh and Kz) at each grid point over the portion of the grid that contains particles.

(Default: PRINT\_TURB\_PARAM\_GRADI = 'N' → don't print Kh and Kz)

### **PRINT PART VEL**

Flag ('Y' or 'N') which controls the printing of all velocities in echo file at each STATUS\_INTERVAL interval.

(Default: PRINT\_PART\_VEL = 'N' → don't print velocities)

### **PRINT TOPOG**

Flag ('Y' or 'N') which controls the printing of topography in echo file at beginning of problem.

(Default: PRINT\_TOPOG = 'Y' → print topography)

### **PROBLEM TITLE**

70-character string specifying the label for the particle plot and other ADPIC output.

### **REFLECT AT MLHGT**

Flag ('Y' or 'N') which determines whether or not particles will be reflected at the top of the mixed layer or to leak out.

Note: This parameter is for use ONLY with DIFF\_METHOD = 'MC1'.

(Default: REFLECT\_AT\_MLHGT = 'Y' → no leakage, i.e., particles are reflected back into the mixed layer)

### **RESTART DATE**

7-character date string (ddMMMy) the restart date of the ADPIC calculation. This restart date/time MUST have a corresponding ADPIC\*.RST and CONC\*.RST files (see RESTART\_WRITE\_INTERVAL description). Also see RESTART\_TIME, which specifies the time.

(RESTART\_DATE = 'unset' is the default, i.e., for a non-restart run)

### **RESTART TIME**

8-character string specifying the restart time (hh:mm:ss) of the ADPIC calculation. This restart date/time MUST have a corresponding dump in the ADPIC\*.RST and CONC\*.RST files (see RESTART\_WRITE\_INTERVAL description). Also see RESTART\_DATE, which specifies the date.

(RESTART\_TIME = 'unset' is the default, i.e., for a non-restart run)

### **RESTART WRITE DATE**

Array (max size = MAX\_RESTART = 10) of 7-character strings specifying the dates (ddmmmyy) in increasing chronological order at which data is output from ADPIC into restart dump files (ADPIC\*.RST and CONC\*.RST). It therefore determines the times at which a problem can be restarted since the dump files contain all of the necessary information to restart the problem. When a restart is initiated, each particle will begin to move from its last location. Since restart files are quite large, the user should be cautious in specifying too many restart files. RESTART\_WRITE\_DATE and RESTART\_WRITE\_TIME override RESTART\_WRITE\_INTERVAL.

### **RESTART WRITE INTERVAL**

13-character string specifying the time interval (ddd::hh:mm:ss) at which data is output from ADPIC into a restart dump files (ADPIC\*.RST and CONC\*.RST). It therefore determines the times at which a problem can later be restarted since the restart file contains all fo the necessary information, i.e., coordinates of all of the particles. When a restart is initiated, each particle will begin to move from its last location. Since restart files are quite large, the user should be cautious in specifying an interval which produces a large number of such files. To specify date/times for restart file writes which are not multiples of RESTART\_WRITE\_INTERVAL, use RESTART\_WRITE\_DATE and RESTART\_WRITE\_TIME; RESTART\_WRITE\_DATE and RESTART\_WRITE\_TIME override RESTART\_WRITE\_INTERVAL

No restart file (RESTART\_WRITE\_INTERVAL = '000::0000') is usually specified for an Initial calculation.

### **RESTART WRITE TIME**

Array (max size = MAX\_RESTART = 10) of 8-character strings specifying the times (hh:mm:ss) in increasing chronological order at which data is output from ADPIC into restart dump files (ADPIC\*.RST and CONC\*.RST). It therefore determines the times at which a problem can be restarted since the dump files contain all of the necessary information to restart the problem. When a restart is initiated, each particle will begin to move from its last location. Since restart files are quite large, the user should be cautious in specifying too many restart files. RESTART\_WRITE\_DATE and RESTART\_WRITE\_TIME override RESTART\_WRITE\_INTERVAL.

### **RUN DURATION**

13-character string specifying the time interval (ddd::hh:mm:ss) for the current ADPIC run. It is relative to RUN\_START\_DATE \ RUN\_START\_TIME if it is not a restart run, and from RESTART\_DATE \ RESTART\_TIME if it is a restart run. Run\_stop\_date and run\_stop\_time be used to specify the length of an ADPIC run in terms of beginning and ending rather than beginning and the duration interval.

### **RUN\_START\_DATE**

7-character string specifying the start date (ddmmmyy) of the ADPIC calculation. Note that this value must remain unchanged from the original run to a restart run.

### **RUN\_START\_TIME**

8-character string specifying the start time (hh:mm:ss) of the ADPIC calculation. Note that this value remains unchanged from the original run to a restart run.

### **RUN\_STOP\_DATE**

7-character string specifying the stop date (ddmmmyy) of the ADPIC calculation. Specifying both RUN\_STOP\_DATE and RUN\_STOP\_TIME determines the total duration of an ADPIC execution. RUN\_STOP\_DATE\RUN\_STOP\_TIME will override RUN\_DURATION.

### **RUN\_STOP\_TIME**

8-character string specifying the stop time (hh:mm:ss) of the ADPIC calculation. Specifying both RUN\_STOP\_DATE and RUN\_STOP\_TIME determines the total duration of an ADPIC execution. RUN\_STOP\_DATE/RUN\_STOP\_TIME will override RUN\_DURATION.

### **SAMP\_CELL\_DELZ\_FRACT**

Floating point value representing the vertical height of the sampling cell expressed as a fraction of DELZ. This allows finer resolution of concentration in the vertical, provided there is a sufficient number of particles in the sampling volume.

(Default: SAMP\_CELL\_DELZ\_FRACT = 1.0)

### **SAMPLING\_HGT**

Array (max size = MAXSBINS = 10) of floating point values specifying the heights (m) above ground at which the output bin contents (determined by SOURCES\_TO\_SAMPLING\_BIN and SAMPLING\_TYPE) are to be calculated. For example:

- a) if inhalation doses or concentrations are desired, a SAMPLING\_HGT value of 1.5m (breathing height) would be used;
- b) if concentrations at stack heights or flight levels are desired, a SAMPLING\_HGT equal to the height of interest would be used;
- c) for ground deposition, a SAMPLING\_HGT of 0.0m would be used.

### **SAMPLING\_INTERVAL**

13-character string specifying the output interval (ddd::hh:mm:ss) at which particle information is written to the CONC file.

(Default: SAMPLING\_INTERVAL = '000::0100' → 1-hour dumps to the CONC file)

### **SAMPLING TYPE**

Array (max size = MAXSBINS = 10) of 6-character strings ('INTAIR', 'INSAIR', 'TOTDEP') representing the type of concentration output desired for the ADPIC output bins:

SAMPLING_TYPE	=	'INTAIR'	→	integrated air concentration
SAMPLING_TYPE	=	'INSAIR'	→	instantaneous air concentration
SAMPLING_TYPE	=	'TOTDEP'	→	total ground deposition }

### **SAMPLING INTERVAL**

13-character string specifying the output interval (ddd::hh:mm:ss) at which particle information is written to the CONC file.

(Default: SAMPLING\_INTERVAL = '000::0100' → 1-hour dumps to the CONC file)

### **SKIP RANDOM COUNT**

Integer number of random numbers to skip in random number sequence before the first random number is used. This parameter is primarily for programmer's use as a debugging tool: use of SKIP\_RANDOM\_COUNT can identify how much variability in results is due to randomness.

### **STATUS INTERVAL**

13-character string specifying the edit interval (ddd::hh:mm:ss) at which particle status information lines are written to the screen and to the echo file.

(Default: STATUS\_INTERVAL = '000::0100' → 1-hour edit interval)

### **SOURCES TO SAMPLING BIN**

2-D array (max ADPIC sources, max sampling bins) of integers (1 to max ADPIC sources) specifying which ADPIC sources are to be associated with which sampling bin. SOURCES\_TO\_SAMPLING\_BIN(1,2) = 3 indicates that sampling bin 2 includes particles from source 3 (it may also include others, this entry just specifies that part of the sampling strategy). The source indices must be used in sequence for a given sampling bin index. See sample ADPIC.NML files in this document for further examples.

(Default: all elements of SOURCES\_TO\_SAMPLING\_BIN default to 0)



### **USE\_EXPL\_CLD\_RISE**

Flag ('Y' or 'N') controlling whether or not the explosive cloud rise option is to be used.

Note: the rest of the parameters for the explosive cloud rise are contained in the namelist  
ADPIC\_EXPL\_CLD

(Default: USE\_EXPL\_CLD\_RISE = 'N'  $\Rightarrow$  do not use explosive cloud rise)

### **USE\_VLST**

Flag ('Y' or 'N') controlling whether the file 'CG-MATHEW.VLST' or 'MEDIC.VLST' is to be used to obtain a list of valid MATVEL/MEDVEL files for this run. 'Y' means use the appropriate vlst file, and 'N' means search for "

'MATVEL\*.BIF' or 'MEDVEL\*.BIF' in the current MODEL\$A\_WINDS directory.

(Default: USE\_VLST = 'N'  $\rightarrow$  search for 'MATVEL\*.BIF' or 'MEDVEL\*.BIF' in the current MODEL\$A\_WINDS directory)

### **WIND\_TYPE**

6-character string ('CG-MATHEW', 'MEDIC') controlling whether MATVEL or MEDVEL files are to be used for input wind velocity data. 'CG-MATHEW' means use MATVEL files, and 'MEDIC' means use MEDVEL files (which have not been conditioned by the CG-MATHEW. program.

(Default: WIND\_TYPE= 'CG-MATHEW'  $\Rightarrow$  use MATVEL files)

### **X\_NEST\_LOC**

Floating point value representing the x-coordinate about which the nested grid structure will be built. This is especially useful if there are multiple source locations and you wish to override the default center (which is the least squares distance from all of the source locations) of the nested grid structure.

(Default: X\_NEST\_LOC = -99999.0 = use default center of all source locations.)

### **X\_WINDOW\_MAX**

Floating point value representing a map coordinate (km) indicating the right-hand X edge of a window to be specified and fixed for outputting dot plots.

(Default: X\_WINDOW\_MAX = 0  $\rightarrow$  use standard windows)

### **X\_WINDOW\_MIN**

Floating point value representing a map coordinate (km) indicating the left-hand X edge of a window to be specified and fixed for outputting dot plots.

(Default: X\_WINDOW\_MIN = 0  $\rightarrow$  use standard windows)

### **Y NEST LOC**

Floating point value representing the y-coordinate about which the nested grid structure will be built. This is especially useful if there are multiple source locations and you wish to override the default center (which is the least squares distance from all of the source locations) of the nested grid structure.

(Default: Y\_NEST\_LOC = -99999.0 = use default center of all source locations.)

### **Y WINDOW MAX**

Floating point value representing a map coordinate (km) indicating the right-hand Y edge of a window to be specified and fixed for outputting dot plots.

(Default: Y\_WINDOW\_MAX = 0 -> use standard windows)

### **Y WINDOW MIN**

Floating point value representing a map coordinate (km) indicating the left-hand Y edge of a window to be specified and fixed for outputting dot plots.

(Default: Y\_WINDOW\_MIN = 0 -> use standard windows)

### **Z WINDOW MAX**

Floating point value representing the grid distance (m) indicating the top of the window to be specified and fixed for outputting dot plots.

(Default: X\_WINDOW\_MAX = 0 -> use standard windows)

### **Z WINDOW MIN**

Floating point value representing the grid distance (m) indicating the bottom of the window to be specified and fixed for outputting dot plots.

### **III.B.9.b. ADPIC\_METPARAMS namelist**

#### **DRIFT VEL METHOD**

Integer (0, 1, 2) specifying the type of drift velocity calculation method to use for Monte Carlo diffusion: DRIFT\_VEL\_METHOD = 0 → Legg and Raupach method

DRIFT\_VEL\_METHOD = 1 → none

DRIFT\_VEL\_METHOD = 2 → W..., T..., and K...method

Note: This parameter is ONLY for use with DIFF\_METHOD = 'MC1 '.

(Default: DRIFT\_VEL\_METHOD = 0 → Legg and Raupach method)

#### **FSIGU**

Floating point factor by which SIGU and SIGV are multiplied in Monte Carlo diffusion.

Note: This parameter is ONLY for use with DIFF\_METHOD = 'MC1 '.

(Default: FSIGU = 1.0)

#### **FSIGW**

Floating point factor by which SIGW is multiplied in Monte Carlo diffusion.

Note: This parameter is ONLY for use with DIFF\_METHOD = 'MC1 '.

(Default: FSIGW = 1.0)

#### **FTLU**

Floating point factor by which TLAGU is multiplied in Monte Carlo diffusion.

Note: This parameter is ONLY for use with DIFF\_METHOD = 'MC1 '.

(Default: FTLU = 1.0)

#### **FTLV**

Floating point factor by which TLAGV is multiplied in Monte Carlo diffusion.

Note: This parameter is ONLY for use with DIFF\_METHOD = 'MC1 '.

(Default: FTLV = 1.0)

#### **FTLW**

Floating point factor by which TLAGW is multiplied in Monte Carlo diffusion.

Note: This parameter is ONLY for use with DIFF\_METHOD = 'MC1 '.

(Default: FTLW = 1.0)

### **GAUSS DIFF MULT X**

Floating point value used as a multiplier to QNSH, which determines how long an ADPIC particle remains a source particle. Also see GAUSS\_DIFF\_MULT\_Z.

(Default: GAUSS\_DIFF\_MULT\_X = 1.0)

### **GAUSS DIFF MULT Z**

Floating point value used as a multiplier to QNSV, which determines how long an ADPIC particle remains a source particle. Also see GAUSS\_DIFF\_MULT\_X.

(Default: GAUSS\_DIFF\_MULT\_Z = 1.0)

### **INTERP VEL WINDS**

8-character string ('NONE', 'LINEAR1', 'LINEAR2', 'GAUSS1', 'GAUSS2') used to invoke time-interpolation of MEDIC or CG-MATHEW wind fields. If the flag is absent from the namelist or set to 'NONE', ADPIC does no interpolation of wind fields and uses wind datasets in a stepwise manner. 'LINEARn' results in linear interpolation between wind field datasets at each model timestep, while 'GAUSSn' results in Gaussian interpolation. For n=1, ADPIC uses dataset start times to define the time intervals for interpolation; for n=2, ADPIC shifts each wind field starting time by half its duration time and uses these shifted times to define the interpolation time intervals. In other words, the winds are true at the beginning of the dataset period for n=1, and at the mid-point of the dataset period for n=2. If calculation time is outside the time range of available wind data, ADPIC uses the wind dataset nearest in time with no interpolation. Parameters in ADPIC which are not strong functions of wind velocities are computed at wind dataset intervals.

(Default: INTERP\_VEL\_WINDS = 'NONE')

### **INV OBUKHOV LEN ML**

Floating point value representing the inverse of the Monin-Obukhov length in the mixed layer. The Monin-Obukhov length (usually referred to as L) is the height above the surface at which mechanical and thermal turbulence are of equal importance. From an estimate of SFC\_ROUGH\_HGT and the atmospheric stability, INV\_OBUKHOV\_LEN\_ML can be found from the nomogram in Figure III.B.2. INV\_OBUKHOV\_LEN\_ML is a time-varying parameter with date/times controlled by MET\_START\_DATE / MET\_START\_TIME.

Warning errors: INV\_OBUKHOV\_LEN\_ML < -0.15

INV\_OBUKHOV\_LEN\_ML > 0.15

Fatal errors: INV\_OBUKHOV\_LEN\_ML < - 1.00

INV\_OBUKHOV\_LEN\_ML > 1.00

INV\_OBUKHOV\_LEN\_ML > 0.0 and STAB\_CLASS < 4

INV\_OBUKHOV\_LEN\_ML < 0.0 and STAB\_CLASS > 4

### INV\_OBUKHOV\_LEN\_SL

Floating point value representing the inverse of the Monin-Obukhov length in the surface layer. The Monin-Obukhov length (usually referred to as  $L$ ) is the height above the surface at which mechanical and thermal turbulence are of equal importance. From an estimate of SFC\_ROUGH\_HGT and the atmospheric stability, INV\_OBUKHOV\_LEN\_SL can be found from the nomogram in Figure III.B.2.

Warning errors: INV\_OBUKHOV\_LEN\_SL < -0.15

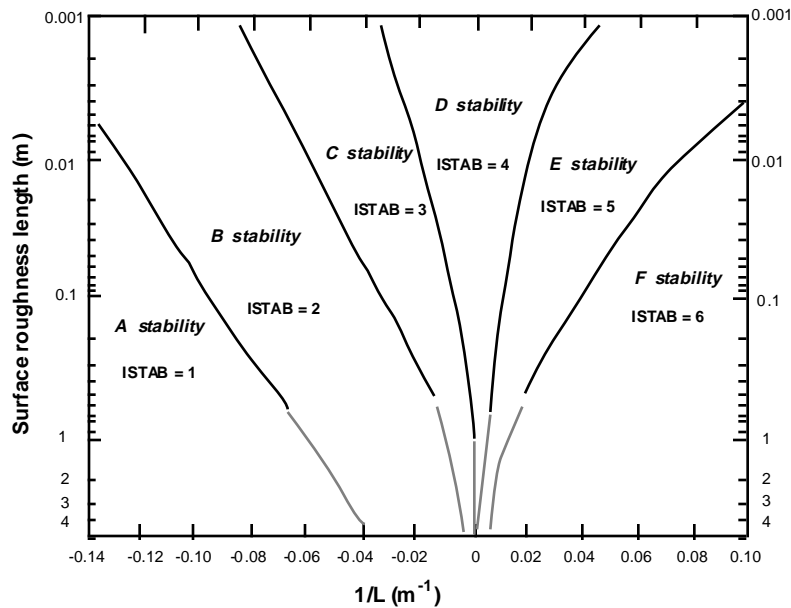
INV\_OBUKHOV\_LEN\_SL > 0.15

Fatal errors: INV\_OBUKHOV\_LEN\_SL < -1.00

INV\_OBUKHOV\_LEN\_SL > 1.00

INV\_OBUKHOV\_LEN\_SL > 0.0 and STAB\_CLASS < 4

INV\_OBUKHOV\_LEN\_SL < 0.0 and STAB\_CLASS > 4



**Figure III.B.2. Relation of Monin-Obukhov  $L$  to Pasquill class and roughness length (Golder, 1972)**

### MAX\_DELZ\_SHIFT

Integer representing the maximum number of cells that a particle can pop up. There is a tendency for particles to impact with topography in complex terrain and become 'stuck'. In order to help alleviate this problem we have introduced a procedure which allows particles to pop up a given number of times to get the particles back out into the flow field. This procedure involves moving the particle up one cell at a time in the vertical until the particle has either cleared the topography or

has reached the allowed limit of vertical shift (MAX\_DELZ\_SHIFT grid cells) or the upper limit of the grid is encountered. If this limit (MAX\_DELZ\_SHIFT) has been reached, the particle is moved only with the velocity component parallel to the topography. This procedure will allow the particle to either move over or around the topography. MAX\_DELZ\_SHIFT specifies the limit of cells a particle is allowed to be shifted vertically.

(Default: MAX\_DELZ\_SHIFT = 1)

### **MAX\_DITHER**

Floating point value specifying the a multiplicative factor of the timestep delta for a given computational cycle, and that product is then used as a time to multiply the maximum particle speed in each direction. When the advection component of the velocity is much less than the diffusion component, the particles will try to line up on grid lines. This is particularly true in the vertical direction. To help break up this alignment we have introduced a small random change (a dither, similar to that in control systems). MAX\_DITHER controls the amount of this small random movement. Since the resulting 'small' distance is in effect a fraction of the maximum movement for the next cycle. By then multiplying by a random number (between -1. and +1.) the correction acts to prevent the alignment on horizontal grid lines.

(Default: MAX\_DITHER = 0.01)

### **MET\_START\_DATE**

Array (max size = MAXTIM = 50) of 7-character strings specifying the dates (ddmmmyy) at which new values of the time-varying met parameters (TURB\_PARAM\_TYPE, STAB\_CLASS, INV\_OBUKHOV\_LEN\_ML, INV\_OBUKHOV\_LEN\_SL, ML\_HGT, SIGMA\_THETA\_FACT, SIGMA\_THETA, SIGMA\_THETA\_HGT, and PRECIP\_SCAVENG\_COEF) take effect. Also see MET\_START\_TIME, which specifies the times.

### **MET\_START\_TIME**

Array (max size = MAXTIM = 50) of 8-character strings specifying the times (hh:mm:ss) at which new values of the time-varying met parameters (TURB\_PARAM\_TYPE, STAB\_CLASS, INV\_OBUKHOV\_LEN\_ML, INV\_OBUKHOV\_LEN\_SL, ML\_HGT, SIGMA\_THETA\_FACT, SIGMA\_THETA, SIGMA\_THETA\_HGT, and PRECIP\_SCAVENG\_COEF) take effect. Also see MET\_START\_DATE, which specifies the dates.

### **MIN\_MASS\_FRACT**

Floating point value representing the factor used to determine whether or not an ADPIC particle should be eliminated due to decreased mass or activity: If mass or activity is less than MIN\_MASS\_FRACT times the original mass or activity of the ADPIC particle, then the particle is eliminated from the problem.

(Default: MIN\_MASS\_FRACT = 0.001)

## **ML\_HGT**

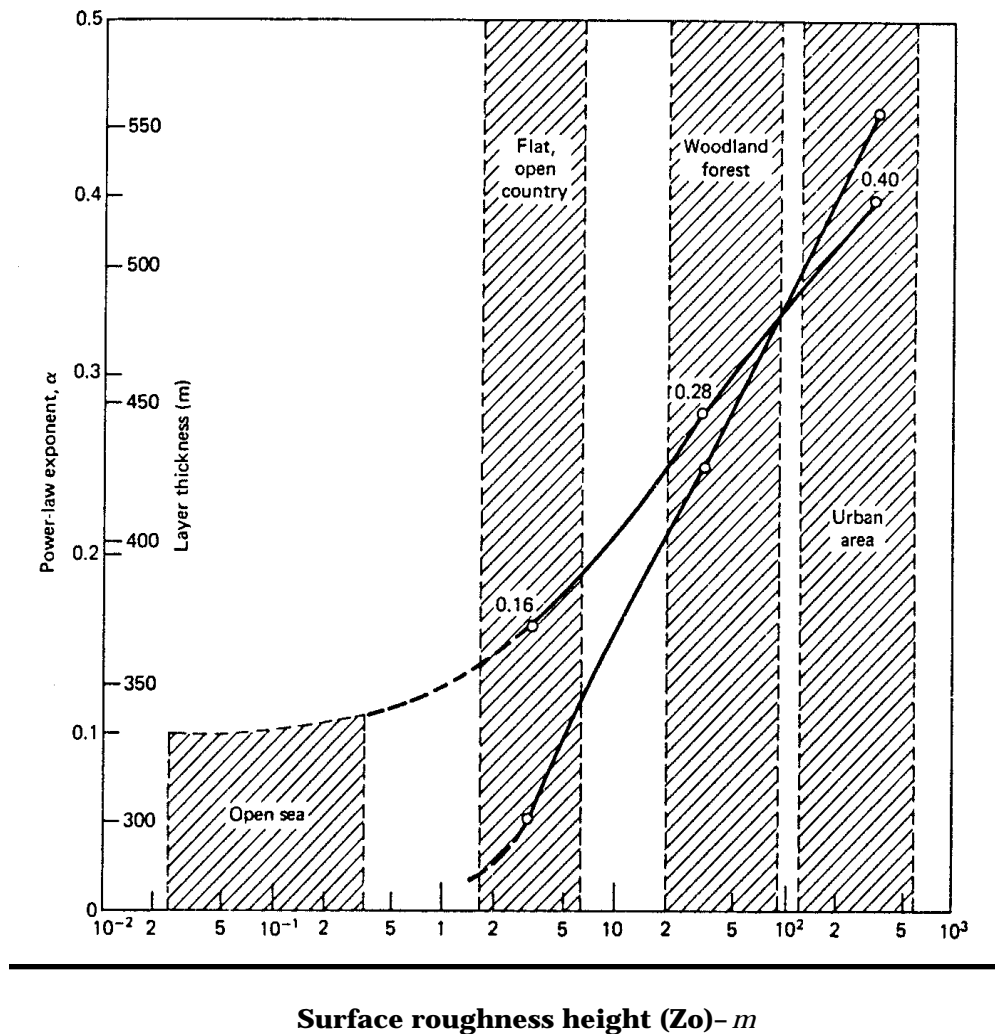
Floating point value representing the height (m) of the mixed layer above terrain. The height of the mixed layer is that demarcation between well mixed, turbid air below, and relatively clear air above. The mixed layer, typically ranging from 300 to 3000 m above terrain, basically indicates the height of the temperature inversion base. ML\_HGT is usually obtained from the radiosonde nearest the source (or area of interest). In the absence of a temperature profile, observations have often shown that the wind will shear rapidly in direction and be accompanied by a speed minima as the sonde passes through the inversion layer.

Warning errors:    ML\_HGT  $\leq$  50.0  
                         ML\_HGT  $\geq$  3000.0  
Fatal errors:        ML\_HGT < DELZ  
                         ML\_HGT  $\geq$  20000.0

## **PWR\_LAW\_EXPNT\_SL**

Floating point value representing the power law exponent for the surface layer. It can be calculated directly from surface-layer wind observations in the vertical using the power law formula given by Eq. (C.1). Alternatively, it can be obtained as a function of surface roughness height and atmospheric stability, as shown in Fig. III.B.1. PWR\_LAW\_EXPNT\_SL is used within the code to extrapolate: (1) wind speeds from their measured height to REF (see below), (2) wind speeds from the SFC\_ROUGH\_HGT to the top of the surface layer above terrain in the parameterized extrapolation procedure (see SL\_HGT below), and (3) wind speeds from the SFC\_ROUGH\_HGT and REF\_HGT in the profile extrapolation procedure (see Section II.C.2.b for details).

Warning errors:    PWR\_LAW\_EXPNT\_SL > 0.5  
Fatal errors:        PWR\_LAW\_EXPNT\_SL < 0.0  
                         PWR\_LAW\_EXPNT\_SL > 10.0



The exponent in the power law and the height of the boundary layer as functions of the roughness height and the type of terrain. From Davenport (1965).

$$Z_o = \text{surface roughness height (m)} \approx h_o/30$$

where  $h_o$  = average height of terrain features (e.g., building, trees, grass)

PWR\_LAW\_EXPNT\_SL is obtained from graph using appropriate  $Z_o$  and the following correction for stability (if necessary):

Stable:	add 0.1
Neutral:	as is
Unstable:	subtract 0.1

**Figure III.B.1 Modeling the planetary boundary layer**



### **PRECIP\_SCAVENG\_COEF**

Floating point value representing the washout coefficient for rainout calculations. PRECIP\_SCAVENG\_COEF must be accompanied by entries for RAIN\_START and RAIN\_STOP if only the first element of PRECIP\_SCAVENG\_COEF is set. Otherwise, PRECIP\_SCAVENG\_COEF is controlled by date/times of the met\_start\_date/met\_start\_time arrays. Suggested values: PRECIP\_SCAVENG\_COEF = 1.0E-3 for light rain, 2.0E-3 for light snow.

### **RAIN\_START**

13-character string specifying the time (ddd::hh:mm:ss) relative to the problem start time (RUN\_START\_TIME) when the rain is to start. This is used in conjunction with RAIN\_STOP (rain stop time) and PRECIP\_SCAVENG\_COEF (washout coefficient), first element only.

### **RAIN\_STOP**

13-character string specifying the time (ddd::hh:mm:ss) relative to the problem stop time (RUN\_START\_TIME) when the rain is to stop. This is used in conjunction with RAIN\_START (rain start time) and PRECIP\_SCAVENG\_COEF (washout coefficient).

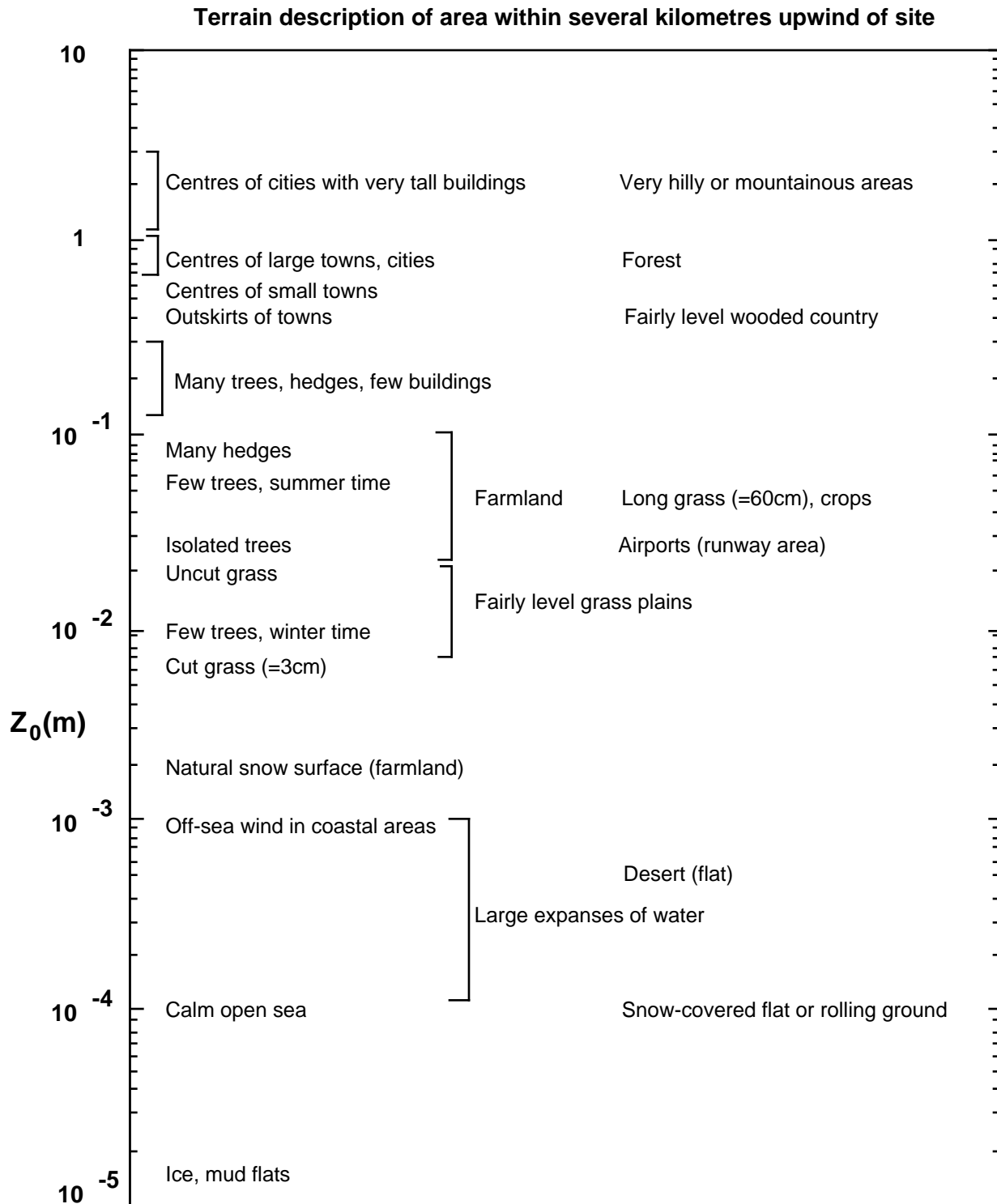
### **SFC\_ROUGH\_HGT**

Floating point value representing the surface roughness height (m). The surface roughness height is approximately the average height of most land-based features (e.g., grass, trees, buildings) divided by 30. The analyst should consider features over the entire gridded area before entering a value. This is a subjective parameter since we are usually dealing with very diverse geographical areas. Figure III.B.3 gives typical values of surface roughness for various terrain types.

Warning errors: SFC\_ROUGH\_HGT  $\geq$  2.0

Fatal errors: SFC\_ROUGH\_HGT  $\leq$  0.0

SFC\_ROUGH\_HGT  $>$  5.0



**Figure III.B.3.  $z_0$  values for typical terrain types (After ESDU 72026, 1972)**

### **SIGMA THETA FACT**

Array (max size = MAXTIM = 50) of floating point values specifying a time-varying factor by which the sigma theta profile generated from the SIGMA\_THETA array of the SIGMA\_THETA\_DATA namelist is multiplied.

### **SIGPHI**

Floating point value specifying the single Sigma-Phi (standard deviation of the vertical wind direction) to be used.

Note: This parameter is ONLY to be used with DIFF\_METHOD ='MC1 ' and TURB\_PARAM\_TYPE=4.

### **STAB CLASS**

Integer (1-6) representing the Pasquill-Gifford stability category for vertical diffusion. It also defines horizontal diffusion if  $\partial\omega$  is absent. There are many ways to determine atmospheric stability. One is the Pasquill-Turner method, which incorporates insolation assessments based on hourly cloud observations. The heat flux at the earth's surface is qualitatively estimated from these cloud observations, and then related to stability on the basis of the station wind speed. This method is similar to the method used to define the stability categories during the original development of the Pasquill-Gifford curves. STAB\_CLASS is a time-varying parameter under control of the MET\_START\_DATE / MET\_START\_TIME arrays

Another method is to use a table, such as the "Relation of Turbulence Types to Weather Conditions" included on page 2 of the MEDIC input form.

A consistent stability for each surface observation is generally made, so the number of values can be quite sizable. Since STAB\_CLASS must represent a composite of the grid-wide stability, the individual stabilities are averaged with a greater weight given to those estimates nearest the source location.

Warning errors: NONE

Fatal errors: STAB\_CLASS < 1  
STAB\_CLASS > 6

### **TIMESTEP FRACT**

Floating point value representing the fraction of the computed time step to use, where the unadjusted time step is the time interval for the fastest particle to travel one cell's distance.

(Default: TIMESTEP\_FRACT = 1.0)

### **TIMESTEP\_FRACT\_MC**

Floating point value representing the fraction of the computed time step to use for the case `diff_method = 'MC1'`, where the unadjusted time step is the time interval for the fastest particle to travel one cell's distance.

(Default: `TIMESTEP_FRACT_MC = 1.0`)

### **TURB\_PARAM\_TYPE**

Integer (1 to 5) specifying the diffusion coefficients to be used. Current possible values of `TURB_PARAM_TYPE` for use with gradient diffusion (`DIFF_METHOD = 'gradi'`) are 1, 2, 4, and 5; and 2, 3, 4, and 5 for Monte Carlo diffusion (`DIFF_METHOD = 'MC1'`).

When stack plume rise is used, the input parameters for the inverse Monin-Obukhov length (`INV_OBUKHOV_LEN_SL`) and the top of the mixed layer (`ML_HGT`) are required for computation of friction velocity for all values of `TURB_PARAM_TYPE`.

<code>TURB_PARAM_TYPE</code>	Diffusion coefficients to be used
1	Gaussian coefficients, Pasquill-Gifford
2	Gaussian coefficients, Idaho Falls MesoDif model
3	Hanna method. (with Monte Carlo diffusion method ONLY) Lagrangian time scale and sigma-w from Steve Hanna.

The following statements apply to both `TURB_PARAM_TYPE`'s 4 and 5: Generalized diffusion parameters are computed for the entire atmospheric boundary layer as a function of wind profile, stability, surface roughness, and mixing layer thickness. The following input parameters must be provided in the `Adpic.nml` file: stability category (`STAB_CLASS`), inverse Monin-Obukhov length (`INV_OBUKHOV_LEN_SL`), top of mixing layer (`ML_HGT`), power law exponent in surface layer (`PWR_LAW_EXPNT_SL`), and surface roughness height (`SFC_ROUGH_HGT`). The vertical diffusion coefficient `Kz` is from Businger, Arya, and Deardorff in the outer mixing layer, and similarity theory in the surface layer. The Idaho Falls MesoDif Gaussian diffusion sigma-y curves are used for sub-grid diffusion. These coefficients result in more diffusion under stable conditions due to meandering effects when compared to the classical Pasquill-Gifford sigma-y curves.

`TURB_PARAM_TYPE = 4` → Sigma-Theta method.  $K_h$ 's obtained from Taylor's Theory using sigma-theta profile (fluctuation of the horizontal wind direction). The sigma-theta profile is input via the `SIGMA_THETA` and `SIGMA_THETA_HGT` arrays in the `namelist sigma_theta_data` of the `ADPIC.NML` namelist file. This diffusion scheme is suitable for continuous and moderately elevated releases within the mixed layer where sigma-theta data are available.

TURB\_PARAM\_TYPE = 5 → Energy dissipation rate (epsilon) method. Kh's obtained from scale-dependent horizontal diffusion using energy dissipation rate, epsilon. This diffusion scheme is suitable for instantaneous surface and elevated releases where horizontal diffusion is based upon the atmospheric energy dissipation rate.

### **U\_CALM**

Floating point value representing the minimum value (m/s) for calm winds in friction velocity calculations. The friction velocity is computed at every grid point, and is dependent on the advection at each of these grid points. An advection velocity of zero will produce a similar friction velocity, which in turn causes the diffusion to go to zero. In order to be able to simulate calm conditions, U\_CALM is used to compute the diffusion whenever the advection has a lower value than U\_CALM.

(Default: U\_CALM = 0.2)

### **U\_MEAN\_HGT**

Floating point value representing the height (m) of mean wind used to compute friction velocity.

(Default: U\_MEAN\_HGT = 10.0)

### **III.B.9.c. ADPIC\_SOURCE namelist**

#### **AMBIENT\_TEMP**

2-D array (MAXTIM = 50, MAXSOURC = 9) of floating point values representing the ambient air temperature (K). as a function of time, for each source . This array is part of the stack plumerise model.

(Default: AMBIENT\_TEMP = 288.)

#### **CENTER\_HGT**

2-D array (MAXTIM = 50, MAXSOURC = 9) of floating point values representing the height (m) above terrain of the source center.

#### **EXIT\_VEL**

Array (max size = MAXSOURC = 9) of floating point values specifying the vertical exit velocity (m/s) of the gas emitted from the stack of each ADPIC source. Used in the plume rise calculation of buoyancy flux and momentum flux. Normal range of values are 3 to 50 m/sec. Any value less than or equal to 2 m/sec is treated as zero. The ratio of horizontal wind velocity at the source location to the vertical exit velocity is used to determine if vertical plume equations or bent-over plume equations are used. If that ratio is less than 0.1, then vertical plume equations are used.

#### **HALFLIFE**

Array (max size = MAXSOURC = 9) of floating point values specifying the radioactive half-life (hours) of each ADPIC source. Use HALFLIFE = 0 for all non-radioactive substances.

#### **HEAT\_EMISSION\_RATE**

Array (max size = MAXSOURC = 9) of floating point values specifying the sensible heat emission rate (MW) for each ADPIC source. This is an optional parameter used (only if STACK\_TEMP is not input) to calculate buoyancy flux in plume rise calculations. Typical values might be 0.01 to 100 MW. This array is part of the stack plumerise model.

#### **INV\_HGT\_AT\_STACK**

2-D array (MAXTIM = 50, MAXSOURC = 9) of floating point values representing the height (m) of strong inversion affecting plume rise. This is an optional parameter; if a value is input, plume rise for source will be limited by this height, but only in cases involving buoyancy plume rise and neutral/unstable stability classes. If a value is not input, the ML\_HGT value serves this function, but only for cases of vertical plumes in neutral/unstable stability classes.

## **PART DENSITY**

Array (max size = MAXSOURC = 9) of floating point values specifying the particle density (kg/m<sup>3</sup>) for each ADPIC source. PART\_DENSITY is usually a value that has been normalized to 1 g/cc = 1000 kg/m<sup>3</sup>, but the actual particle density may also be used. Note that the PART\_DENSITY chosen should be consistent with the values for PART\_DIAM\_MEDIAN, PART\_DIAM\_MIN, and PART\_DIAM\_MAX.

(Default: PART\_DENSITY = 1000. → 1 g/cc)

## **PART DIAM MAX**

Array (max size = MAXSOURCE = 9) of floating point values specifying the largest particle diameter ( $\mu\text{m}$ ) of the effluent. It is usually taken to be two orders of magnitude greater than PART\_DIAM\_MEDIAN. The terminal fall velocities for those particles falling outside the realm of validity of Stoke's Law, i.e., Reynolds number greater than one, are calculated using an algorithm derived from McDonald's computational aid for the terminal fall velocity of spheres. See Section II.B.4.d for more information. If PART\_DIAM\_MEDIAN is set to zero, no entry is required for PART\_DIAM\_MAX. Typical values are:

PART\_DIAM\_MAX = 200.0   Æ  Pu HE detonation, based upon CS-I data

PART\_DIAM\_MAX = 200.0   Æ  Pu fire, based upon Rocky Flats data

## **PART DIAM MEDIAN**

Array (max size = MAXSOURCE = 9) of floating point values specifying the effluent particle size median diameter ( $\mu\text{m}$ ) for each ADPIC source. If there is no particular range of particle sizes, use PART\_DIAM\_MEDIAN of zero, which will result in a zero gravitational settling velocity. Typical values are:

PART\_DIAM\_MEDIAN = 40.0   → Pu HE detonation, based upon CS-I data

PART\_DIAM\_MEDIAN = 0.6   → Pu fire, based upon Rocky Flats data

PART\_DIAM\_MEDIAN = 0.0   → passive pollutant (gases, smoke,...)

## **PART DIAM MIN**

Array (max size = MAXSOURCE = 9) of floating point values specifying the smallest particle diameter ( $\mu\text{m}$ ) of the effluent. It is usually taken to be two orders of magnitude less than PART\_DIAM\_MEDIAN. If PART\_DIAM\_MEDIAN is set to zero, no entry is required for PART\_DIAM\_MIN. Typical values are:

PART\_DIAM\_MIN = 0.2   → Pu HE detonation, based upon CS-I data

PART\_DIAM\_MIN = 0.2   → Pu fire, based upon Rocky Flats data

### **PART DIAM SGD**

Array (max size = MAXSOURCE = 9) of floating point values specifying the standard geometric deviation of the log-normal particle size distribution for each ADPIC source, and is defined as the following ratio:

$$\frac{\text{radius at 50\%}}{\text{radius at 16\%}} \quad \text{or} \quad \frac{\text{radius at 84\%}}{\text{radius at 50\%}}$$

on a log-normal frequency distribution plot. If PART\_DIAM\_MEDIAN is set to zero, no entry is required for PART\_DIAM\_SGD. Typical values are:

PART\_DIAM\_SGD = 5.71 → Pu HE detonation, based upon CS-I data

PART\_DIAM\_SGD = 3.0 → Pu fire, based upon Rocky Flats data

### **RELEASE START** (recommend use of SOURCE\_START\_DATE and SOURCE\_START\_TIME)

Array (max size = MAXSOURCE = 9) of 13-character time interval strings (ddd::hh:mm:ss) specifying the release start time relative to the run start time (RUN\_START\_DATE \ RUN\_START\_TIME for non-restart, RESTART\_DATE \ RESTART\_TIME for restart) for each ADPIC source. Use of SOURCE\_START\_DATE and SOURCE\_START\_TIME will override use of RELEASE\_START. Use of SOURCE\_START\_DATE and SOURCE\_START\_TIME instead of RELEASE\_START is recommended because they are absolute times and allow for time-varying source rates.

### **RELEASE STOP** (recommend use of SOURCE\_START\_DATE and SOURCE\_STOP\_TIME)

Array (max size = MAXSOURCE = 9) of 13-character time interval strings(ddd::hh:mm:ss) specifying the release stop time relative to the run start time (RUN\_START\_DATE \ RUN\_START\_TIME for non-restart, RESTART\_DATE \ RESTART\_TIME for restart) for each ADPIC source. RELEASE\_STOP will override RUN\_DURATION. Use of SOURCE\_START\_DATE and SOURCE\_START\_TIME will override use of RELEASE\_STOP. Use of SOURCE\_START\_DATE and SOURCE\_START\_TIME instead of RELEASE\_STOP is recommended because they are absolute times and allow for time-varying source rates.

### **SFC DEP HGT**

Floating point value representing the height (m) above ground level used for dry deposition calculations. SFC\_DEP\_VEL decreases linearly with height and goes to zero at the height specified by SFC\_DEP\_HGT.

(Default: SFC\_DEP\_HGT = DELZ from TOPOG.NML)

### **SFC DEP VEL**

Array (max size = MAXSOURCE = 9) of floating point values specifying the surface deposition velocity (m/s) of the effluent. This is not a gravitational settling velocity, but rather a measure of the affinity of the material for the ground. SFC\_DEP\_VEL is a function of particle size, atmospheric



stability, and surface roughness height. The surface deposition velocity is the downward particle flux divided by the particle concentration--so the units are m/sec. SFC\_DEP\_VEL is only used in the bottom grid cell to deposit particles onto the ground, and is weighted according to the distance from the bottom of the cell (i.e., zero at the top of the cell to full value at the bottom). The only mechanisms by which particles or a portion of the particle activity can be removed from the atmosphere and deposited on the ground are via SFC\_DEP\_VEL, rainout, and gravitational settling. Experimental data have shown that SFC\_DEP\_VEL usually lies between 0.001 and 0.05 m/s:

<u>Particulate Effluent</u>	<u>SFC DEP VEL (m/s)</u>
Pu-238, Pu-239	0.001- 0.01
SO <sub>2</sub> , Ru	0.01- 0.03 (0.028)
Cs-137, Sr-90	0.001- 0.002
I-131	0.003
Inert particles	0.001- 0.002
HTO	0.005- 0.008

### **SIGX**

Array (max size = MAXSOURC = 9) of floating point values specifying the horizontal standard deviation (m) in the X-direction (East-West) of the initial stabilized source geometry for each ADPIC source.

### **SIGY**

Array (max size = MAXSOURC = 9) of floating point values specifying the horizontal standard deviation (m) in the Y-direction (North-South) of the initial stabilized source geometry for each ADPIC source.

### **SIGZ**

Array (max size = MAXSOURC = 9) of floating point values specifying the vertical standard deviation (m) of the initial stabilized source geometry for each ADPIC source.

### **SOURCE RATE**

2-D array (MAXTIM = 50, MAXSOURC = 9) specifying the floating point values for the source rate. The actual units of the bin contents to be plotted by PLCNT are determined by a combination of SOURCE\_RATE, DCON (a PLCNT parameter), and the ADPIC internal units. See the description for DCON in the PLCNT section of this User's Guide for a more complete discussion of the interrelation of these parameters. A source rate which is constant can be set by inputting the first element of SOURCE\_RATE for each source. In this case, values of SOURCE\_START\_TIME and SOURCE\_START\_DATE are not required provided RELEASE\_START and RELEASE\_STOP are also set appropriate.y.

### **SOURCE START DATE**

Array (max size = MAXTIM = 50) of 7-character date strings (ddMMMyy) specifying the dates at which new values of the time-varying source parameters (SOURCE\_RATE and CENTER\_HGT) take effect. Also see SOURCE\_START\_TIME, which specifies the times

### **SOURCE START TIME**

Array (max size = MAXTIM = 50) of 8-character time strings (hh:mm:ss) specifying the times at which new values of the time-varying source parameters (SOURCE\_RATE and CENTER\_HGT) take effect. Also see SOURCE\_START\_DATE, which specifies the dates.

### **SPECIES**

Array (max size = MAXSOURC = 9) of 10-character strings specifying the name of the effluent for each ADPIC source.

### **STACK RADIUS**

Array (max size = MAXSOURC = 9) of floating point values specifying the internal stack radius (m) for each ADPIC source. This is used to calculate buoyancy flux and momentum flux for plume rise calculations. It is NOT used to specify the initial source geometry. The parameters ZPOS\_CUTOFF, ZNEG\_CUTOFF, XPOS\_CUTOFF, XNEG\_CUTOFF, YPOS\_CUTOFF, YNEG\_CUTOFF, SIGX, SIGY, and SIGZ are always used for this and should be consistent with STACK\_RADIUS. Typical values are 1 to 5 m.

### **STACK TEMP**

Array (max size = MAXSOURC = 9) of values specifying the temperature (degrees K) of the gas emitted from the stack of each ADPIC source. Used to calculate buoyancy flux for plume rise calculations.

### **TOTPART**

Array (max size = MAXSOURC = 9) of integers specifying the total number of particles to be generated for each ADPIC source. TOTPART is used to calculate the particle release rate. Although ADPIC allows a total maximum of 20,000 particles from all sources at any given time, a continuous release with particle depletion (via deposition, advection, and upward diffusion) can easily release more than 20,000 particles without exceeding this limit at any one time. However, if this maximum is exceeded, ADPIC will redistribute the mass or activity evenly among the 20,000 particles. Particles will be removed from the grid via deposition, advection, or upward diffusion.

(Recommendation: TOTPART = 3000 for multi-species, TOTPART = 5000 for single species)

### **USE PLUME RISE**

Array (max size = MAXSOURC = 9) of integer flags ('Y' or 'N') indicating whether plume rise should be calculated for a specific source. If other plume rise parameters are present, but USE\_PLUME\_RISE = 'N', then the other parameters will be ignored.

(Default: USE\_PLUME\_RISE = 'N' → no plume rise)

### **VERT TEMP GRAD**

Array (MAXTIM = 50, MAXSOURC = 9) of floating point values representing vertical temperature gradient (K/m) for use in plume rise calculations. The time varying dimension corresponds to the same index in SOURCE\_START\_DATE and SOURCE\_START\_TIME. Used to calculate the Brunt-Vaisalla frequency. Defaults are provided in ADPIC for each stability class if a value is not input. (Note: The adiabatic, neutral temperature gradient is -0.0098 K/m. The standard atmosphere temperature gradient is -0.0065 K/m)

### **XLOC**

Array (max size = MAXSOURC = 9) of floating point values specifying the Model x coordinate (km) of the source center. Note: Both XLOC and YLOC may not be closer than 3 grid cells from the edges of the ADPIC grid.

### **XNEG CUTOFF**

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the negative X-direction (i.e., to the West or “left”), referenced to the source center (XLOC) for each ADPIC source.

(Note: since this is distance is reference to the source center, XNEG\_CUTOFF should have a negative value.)

### **XPOS CUTOFF**

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the positive X-direction (i.e., to the East or “right”), referenced to the source center (XLOC) for each ADPIC source.

### **YLOC**

Array (max size = MAXSOURC = 9) of floating point values specifying the Model y coordinate (km) of the source center. Note: Both XLOC and YLOC must be no closer than 3 grid cells from the edges of the ADPIC grid.

### **YNEG\_CUTOFF**

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the negative Y-direction (i.e., to the South or “left”), referenced to the source center (YLOC) for each ADPIC source.

(Note: since this is distance is reference to the source center, YNEG\_CUTOFF should have a negative value.)

### **YPOS\_CUTOFF**

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the positive Y-direction (i.e., to the North or “right”), referenced to the source center (YLOC) for each ADPIC source.

### **ZNEG\_CUTOFF**

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the negative Z-direction (i.e., downward or to the “bottom”), referenced to the source center height (CENTER\_HGT) for each ADPIC source.

(Note: since this is distance is reference to the source center height, ZNEG\_CUTOFF should have a negative value.)

### **ZPOS\_CUTOFF**

Array (max size = MAXSOURC = 9) of floating point values specifying the radial cutoff of the initial stabilized source geometry in the positive Z-direction (i.e., upward or to the “top”), referenced to the source center height (CENTER\_HGT) for each ADPIC source.

### III.B.9.d. SIGMA\_THETA\_DATA namelist

#### SIGMA\_THETA

Array (MAXTIM = 50 by MAXLEVEL\_ALL = 35) of floating point values specifying sigma theta values (the standard deviation of wind direction, expressed in degrees). The first index matches entries in MET\_START\_DATE/MET\_START\_TIME while the last index matches the corresponding index of the array SIGMA\_THETA\_HGT. Sigma\_theta is only used for TURB\_PARAM\_TYPE = 4, which selects the k-theory model. The scheme for determining the values of SIGMA\_THETA depending upon the quantity of input data is given in the table below:

	<b>SIGMA_THETA</b>		
<b>Height</b>	<i>no data specified</i>	<i>single data point</i>	<i>two or more data points</i>
below 10m	same as at 10m	same as at 10m, unless explicitly specified	same as at 10m, unless explicitly specified
10m	A stability: SIGMA_THETA = 25 B stability: SIGMA_THETA = 20 C stability: SIGMA_THETA = 15 D stability: SIGMA_THETA = 10 E stability: SIGMA_THETA = 5 F stability: SIGMA_THETA = 2.5	use power law to correct the given sigma theta to a reference height of 10m, if not already given at 10m	linear interpolation scheme
0.02 ML_HGT 0.04 ML_HGT 0.07 ML_HGT 0.10 ML_HGT	use power law		(Warning to be issued if positive gradient of 5 deg / 100m exceeded)
0.2 ML_HGT 0.4 ML_HGT 0.7 ML_HGT 1.0 ML_HGT	<b>A-D stability:</b> Continue with power law up to 0.9 ML_HGT, then taper to 1 deg at 1.1 ML_HGT. <b>E&amp;F stability:</b> Linear interpolation between value at 0.1 ML_HGT and 1 deg at ML_HGT		(Assumes SIGMA_THETA of 1 deg at ML_HGT and above if not explicitly specified and not implied by the slope of the highest two outer points)

#### SIGMA\_THETA\_HGT

Array (MAXTIM = 50 by MAXLEVEL\_ALL = 35) of floating point values specifying the height (m) above terrain of the SIGMA\_THETA array values. SIGMA\_THETA and SIGMA\_THETA\_HGT must always be entered as pairs. The interpolation algorithm assumes that heights are ordered from lowest to highest.

### **III.B.9.e. ADPIC\_EXPL\_CLD namelist**

The ADPIC\_EXPL\_CLD namelist directs the explosive cloud rise model in ADPIC. However, whether or not variables are set using this namelist, the explosive cloud rise calculation will only be made if the flag USE\_EXPL\_CLD\_RISE is set in the previous ADPIC.NML namelist ADPIC\_CONTROL

#### **DIF IN EXPL CLD**

Character ('Y' or 'N') specifying whether or not diffusion is permitted for particles within the explosive cloud.

(Default: DIF\_IN\_EXPL\_CLD = 'Y' → diffusion is permitted within the explosive cloud)

#### **DTDZ1**

Floating point value representing the lapse rate from ground to elevation of lowest layer (specified by parameter ZI). Degrees C / m.

#### **DTDZ2**

Floating point value representing the lapse rate for elevations higher than the lowest layer, ZI. Degrees C / m.

#### **HEAMT**

Array (max size = number of explosive cloud sources, currently limited to 1) of floating point values representing the mass (# of TNT equivalent) of high explosive contributing to the explosion for each explosive cloud source.

#### **HEATDET**

Array (max size = number of explosive cloud sources, currently limited to 1) of floating point values representing the heat of detonation (J/kg) for each explosive cloud source. For high explosive detonation, use HEATDET = 5.52E6 J/kg.

#### **PLOT\_EXPL\_CLD**

Integer specifying the time interval (sec) between ADPIC particle plots during the dynamic explosive cloud rise period of the ADPIC calculation. The possible range is from 1 through 60; if an out-of-range value is specified, the default value is used.

(Default: PLOT\_EXPL\_CLD = 60 → 60 second interval between particle plots)

### **PRINT EXPL CLD**

Flag ('Y'/'N') which controls creation and printing of the output text file EXPL\_CLD.PRT.

(Default: 'N')

### **PRINT TYPE EXPL CLD**

5-character string ('SHORT'/'FULL') which controls the amount of output to the output text file EXPL\_CLD.PRT.

(Default: 'SHORT')

### **REL HUM**

Floating point value ( $0 < \text{REL\_HUM} < 1.0$ ) representing the relative humidity (fraction) at the surface in the explosive cloud source area.

### **SIG OVER R**

Floating point value which is a multiplier which converts the explosive cloud radius into the initial sigma. It is applied in all three directions, i.e., the initial sigma in each direction is

$$\text{sig} = \text{sig\_over\_r} * \text{r\_expl\_cld}$$

(Default: SIG\_OVER\_R = 10.)

### **SRC EXPL CLD**

Double-dimensioned array (max size = ADPIC source counter by the number of explosive cloud sources, currently limited to 1) of integers (values = ADPIC source numbers) specifying which ADPIC source (1 to MAXSOURC) goes into which explosive cloud source. The first index is a counter for which of the MAXSOURC possible ADPIC sources will be used with the specified explosive cloud source (the second index). For example, "SRC\_EXPL\_CLD(1,1) = 2 3 6" indicates that ADPIC sources #2, #3, and #6 will be contained in a single explosive cloud source #1. NOTE: MULTIPLE EXPLOSIVE CLOUD MODELING IS NOT YET IMPLEMENTED!!!!

### **SRC PRESS**

Floating point value representing the atmospheric pressure (mb) at the height of the initial explosive cloud source.

### **TA0**

Floating point value representing the ground level temperature, degrees C.

### **TEMP\_PROFILE**

Array (max size = MAX\_EXPL\_CLD\_PROF = 30), giving an air temperature profile (C) at various levels in the atmosphere, at heights specified by TEMP\_PROFILE\_HGT).

### **TEMP\_PROFILE\_HGT**

Array (max size = MAX\_EXPL\_CLD\_PROF = 30), representing the height (m) above ground level for the temperature profile given in the array TEMP\_PROFILE.

### **UAEXP**

Floating point value representing the wind power-law exponent.

### **UAREF**

Floating point value representing wind speed (m/s) at the reference height UAZREF.

### **UAZREF**

Floating point value representing the wind speed reference height (m).

### **VEL\_COUPL**

Floating point value representing the velocity coupling coefficient. The suggested value of “VEL\_COUPL = 0.35” was determined from the Clean Slate I dataset of the Roller Coaster experiment.

### **ZI**

Floating point value representing the height (m) of the lowest layer.

### **ZINIT**

Array (size = number of explosive cloud sources, currently limited to 1) of floating point values representing the initial height (m) above ground level of the center of each explosive cloud source.



## Sample ADPIC.NML Namelist Input File

<b>\$adpic_control</b>	
PROBLEM_TITLE = 'Jul 91: Run1G - Stability & src hgt'	<i>namelist title title for plots</i>
RUN_START_DATE = '01APR85'	<i>start of run</i>
RUN_START_TIME = '00:00'	
RUN_DURATION = '00:04:00'	<i>duration of run</i>
SAMPLING_INTERVAL = '00:01:00'	<i>interval for CONC files</i>
RESTART_WRITE_INTERVAL = '00:00:00'	<i>interval for .RST files</i>
PLOT_INTERVAL = '00:01:00'	<i>dot plot interval</i>
STATUS_INTERVAL = '00:00:15'	<i>screen write interval</i>
INITIAL_TIME_STEP = 30	<i>initial time step</i>
DIFF_METHOD = 'GRAD'	<i>diffusion method</i>
SOURCES_TO_SAMPLING_BIN(1,1) = 1	
SOURCES_TO_SAMPLING_BIN(1,2) = 1	
SOURCES_TO_SAMPLING_BIN(1,3) = 1	
SOURCES_TO_SAMPLING_BIN(1,4) = 2	
SOURCES_TO_SAMPLING_BIN(1,5) = 2	<i>bin arrangement</i>
SOURCES_TO_SAMPLING_BIN(1,6) = 2	
SOURCES_TO_SAMPLING_BIN(1,7) = 3	
SOURCES_TO_SAMPLING_BIN(1,8) = 3	
SOURCES_TO_SAMPLING_BIN(1,9) = 3	
SOURCES_TO_SAMPLING_BIN(1,10) = 1 2 3	
SAMPLING_TYPE = 'INTAIR' 'INSAIR' 'TOTDEP' 'INTAIR' 'INSAIR' 'TOTDEP' 'INTAIR' 'INSAIR' 'TOTDEP' 'INTAIR'	<i>types of sampling bins</i>
SAMPLING_HGT = 1.5 1.5 0.0 1.5 1.5 0.0 1.5 1.5 0.0 75.	<i>hgts of sampling bins</i>
PARTPOS = 'P'	<i>for dots on terrain</i>
Send	<i>end of namelist</i>
<b>\$ADPIC_METPARAMS</b>	<i>namelist title</i>
MET_START_DATE = '01APR85' '01APR85' '01APR85' '01APR85'	<i>time for met params</i>
MET_START_TIME = '00:00' '01:00' '02:00' '03:00'	
INV_OBUKHOV_LEN_SL = 0.0 -0.06 0.0 0.06	
PWR_LAW_EXPNT_SL = 0.17 0.13 0.17 030	
INV_OBUKHOV_LEN_ML = -99999.0 -99999.0 -99999.0 -99999.0	
ML_HGT = 500 750 500 250	
TURB_PARAM_TYPE = 5 5 5 5	
STAB_CLASS = 4 2 4 6	
SFC_ROUGH_HGT = 0.10	
Send	<i>end of namelist</i>

<b>\$adpic_source</b>					<i>namelist title</i>
SPECIES	=	'GND'	'ELEVATED'	'COLUMN'	<i>species names</i>
SIGX	=	1.0	1.0	4.0	<i>std deviation in x</i>
SIGY	=	1.0	1.0	4.0	<i>std deviation in y</i>
SIGZ	=	1.0	1.0	40.0	<i>std deviation in z</i>
XPOS_CUTOFF	=	2.5	2.5	10.	<i>x cutoff to right</i>
XNEG_CUTOFF	=	-2.5	-2.5	-10.	<i>x cutoff to left</i>
YPOS_CUTOFF	=	2.5	2.5	10.	<i>y cutoff to forward</i>
YNEG_CUTOFF	=	-2.5	-2.5	-10.	<i>y cutoff to backward</i>
ZPOS_CUTOFF	=	2.5	2.5	80.	<i>z cutoff to top</i>
ZNEG_CUTOFF	=	-0.	-2.5	-120.	<i>z cutoff to bottom</i>
XLOC	=	105.	105.	105.	<i>x coord of src location</i>
YLOC	=	135.	135.	135.	<i>y coord of src location</i>
TOTPART	=	6000	6000	6000	<i>total # of particles</i>
HALFLIFE	=	0.0	0.0	0.0	<i>decay half life</i>
SFC_DEP_VEL	=	0.01	0.01	0.01	<i>deposition velocity</i>
PART_DIAM_MEDIAN	=	0.3	0.3	0.3	<i>particle median diam.</i>
PART_DIAM_SGD	=	3.0	3.0	3.0	<i>particle std geom dev</i>
PART_DIAM_MIN	=	0.1	0.1	0.1	<i>particle min. diameter</i>
PART_DIAM_MAX	=	100.	100.	100.	<i>particle max. diameter</i>
PART_DENSITY	=	1000.	1000.	1000.	<i>particle density</i>
USE_PLUME_RISE	=	'N'	'N'	'N'	<i>plume rise flag</i>
SOURCE_START_DATE	=	'01APR85'	'01APR85'	'01APR85'	<i>start for src rates</i>
SOURCE_START_TIME	=	'00:00'	'02:00'	'05:00'	
SOURCE_RATE(1,1)	=	1.0	0.0	0.0	<i>rates for 1st species</i>
SOURCE_RATE(1,2)	=	1.0	0.0	0.0	<i>rates for 2nd species</i>
SOURCE_RATE(1,3)	=	0.0	1.0	0.0	<i>rates for 3rd species</i>
CENTER_HGT(1,1)	=	0.	100.	120.	<i>hgts for 1st species</i>
CENTER_HGT(1,2)	=	0.	100.	120.	<i>hgts for 2nd species</i>
CENTER_HGT(1,3)	=	0.	100.	120.	<i>hgts for 3rd species</i>
\$end					<i>end of namelist</i>
<b>\$adpic_expl_cld</b>					<i>namelist title</i>
ZINIT	=				
HE_AMT	=				
HEATDET	=				
REL_HUM	=	-99999.0			
SRC_PRESS	=	-99999.0			
TEMP_PROFILE_HGT	=				
TEMP_PROFILE	=				
\$SEND					<i>end of namelist</i>

### **III.B.10. ADPIC.CLST**

PLCNT can optionally read a list of fully qualified, and therefore unique, CONC\*\*\*.BIF filenames from ADPIC.CLST. This file is produced by ADPIC and contains the list of CONC\*\*\*.BIF files created by the most recent ADPIC run. This is an ASCII file that can be edited by an assessor and provides a mechanism to easily handle multiple CONC groups in a single directory by allowing the user to maintain a series of ADPIC.CLST files (under different names with the desired file being copied to ADPIC.CLST as needed). In the absence of an ADPIC.CLST file, PLCNT processes the entire set of unique CONC\*\*\*.BIF qualifiers in the appropriate directory, taking the highest version number associated with each qualifier.

### **III.B.11. CONC\_ANALYSIS.NML**

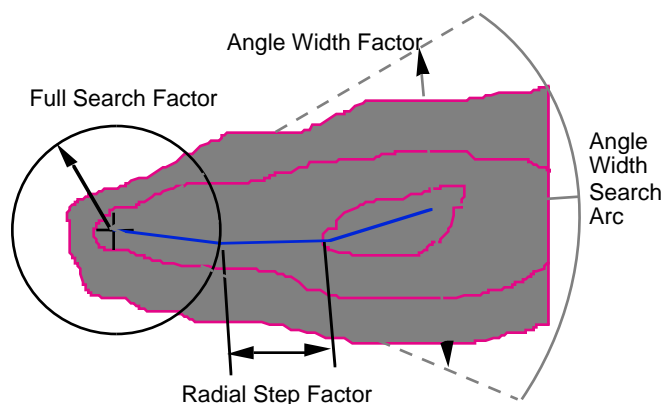
CONC\_ANALYSIS.NML is the input file that directs the execution of both PLCNT and TIMEHIS. In PLCNT it supports control of the types of plots produced and the times when they are produced, the dose and final bin processing, the contour levels and numerous details of the plot format and labeling. TIMEHIS is supported by the same dose and final bin processing and the same time controls as PLCNT but for the interpolation of point concentration values.

The following is a list of the CONC\_ANALYSIS.NML variables grouped alphabetically within namelist blocks: CONC\_ANALYSIS\_CONTROL, PLCNT\_GRAPHICS\_PARAMS, PLCNT\_LABELING\_PARAMS, and TIMEHIS\_PARAMS. The first namelist block is used by both codes, the next two are used only by PLCNT, and the last block only by TIMEHIS. This order must be maintained by all namelist blocks present in the file, although unused blocks can be omitted.

### III.B.11a. CONC\_ANALYSIS\_CONTROL namelist

#### **ANGLE WIDTH FACTOR**

Floating point factor applied to the horizontal (x) grid cell size for the centerline capability. This parameter controls the distance tangent to the previous maximum over which the centerline search is performed. A line of the specified length is evaluated perpendicularly to a line from the source to the previous centerline point at that centerline point. The centerline function then calculates an arc tangential to this line and the search for the next centerline value is determined along this arc:



This parameter takes over at radii beyond the FULL\_SEARCH\_FACTOR. This feature works best with a plume that moves directly away from the source. It can have problems with a meandering plume, especially a plume that ends up oriented radially around the source. To avoid the zig-zags, it is best to leave this value low (e.g., 2 to 4); however if the value is too low the maximum along an arc may be missed.

(Default: ANGLE\_WIDTH\_FACTOR = 3.0)

#### **CENTERLINE\_SOURCE**

Integer (1 to MAXSOURC) representing the ADPIC source whose location is to be used as a starting point for the centerline processing. This is the only required input for the centerline capability; if it is missing, then the centerline function is not performed.

(Default: CENTERLINE\_SOURCE = missing)

#### **CONC\_RESTART\_DATE**

7-character string (ddmmmyy) specifying the restart date of an ADPIC\*.RST file. PLCNT will read the corresponding CONC\*.RST file and accumulate concentration data from all CONC\*.BIF files with times greater than the restart time and generate contour plots. Note that there must be at least one CONC\*.BIF file with a time greater than the restart time. This feature enables the user to not have to have all of the CONC\*.BIF files in a single directory in order to make a PLCNT run which needed integrated air concentration data.

## **CONC RESTART TIME**

8-character string (hh:mm:ss) specifying the restart time of an ADPIC\*.RST file. PLCNT will read the corresponding CONC\*.RST file and accumulate concentration data from all CONC\*.BIF files with times greater than the restart time and generate contour plots. Note that there must be at least one CONC\*.BIF file with a time greater than the restart time. This feature enables the user to not have to have all of the CONC\*.BIF files in a single directory in order to make a PLCNT run which needed integrated air concentration data.

## **DCON**

Array (max size = MAXFBINS = 30) of values, each of which act as a numerical multiplier to the concentrations calculated by ADPIC for a dose bin. The size of the array is required to be equal to the number of dose bins (maximum of 30) specified in SAMPLING\_TO\_DOSE\_BIN. The DCON parameter must either be specified completely or be omitted, in which case the concentrations in the dose bins remain unchanged. The actual units of the bin contents to be plotted by PLCNT are determined by a combination of SOURCE\_RATE (an ADPIC.NML parameter), DCON, and the ADPIC model units:

$$\text{actual units} = (\text{SOURCE\_RATE}) \times (\text{DCON}) \times (\text{ADPIC model units})$$

There are several possible combinations involving SOURCE\_RATE and DCON; below are four of the most common:

- 1) SOURCE\_RATE = actual source rate (e.g., g/s or Ci/s),  
DCON = dose conversion factor (dcf)
- 2) SOURCE\_RATE = unit source rate in 1/s,  
DCON = dcf and the actual amount (e.g., g or Ci)
- 3) SOURCE\_RATE = unit amount (instead of a rate),  
DCON = dcf  $\times$  actual source rate (e.g., g/s or Ci/s)
- 4) SOURCE\_RATE = dcf  $\times$  actual source rate (e.g., g/s or Ci/s),  
DCON = 1

Combination #3 is the one used by the model parameters system. Combination #4 is currently necessary when dose plots for multi-nuclides are desired.

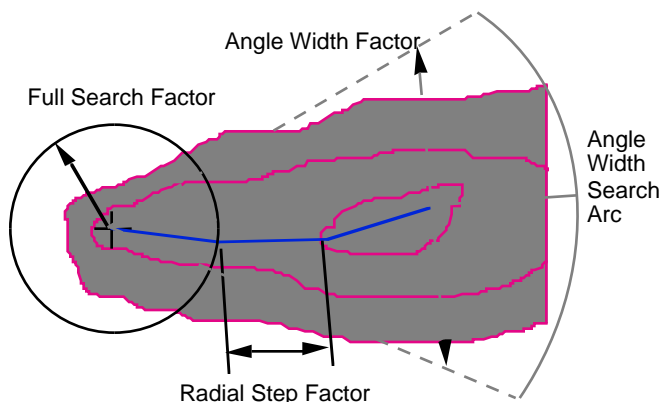
## **DOSE TO FINAL BIN**

2-D array (max size = MAXDBINS, MAXFBINS = 30) of integers (values = dose bin numbers) specifying which dose bin numbers will be combined into final bin  $n$ , i.e., the DOSE TO FINAL bin correlation. This allows for specification of the combination of dose bins to be delayed until PLCNT/TIMEHIS, as well as allowing for the re-use of the dose bins. (Also see SAMPLING\_TO\_DOSE\_BIN.)

## **FULL SEARCH FACTOR**

Floating point factor applied to the horizontal (x) grid cell size for the centerline capability. This parameter controls the radial distance over which to perform a search from SECTOR\_ANGLE\_MIN

counterclockwise to SECTOR\_ANGLE\_MAX (usually 0 to 360 degrees) for a concentration maximum; thus the ANGLE\_WIDTH\_FACTOR is ignored within this radius:



This search was designed to be the initial search pattern to find out which way the plume is going. It is not recommended to make this value very high, since in a splotchy, well spread out plume it will cause a zig-zag centerline that will follow individual high points across the map. A value of 1 to 3 should be adequate for simple plumes. In a complex plume case where the plume initially extends one direction, then breaks up, and later extends in a different direction, you may need to use a larger FULL\_SEARCH\_FACTOR to find the beginning of the later extension.

(Default: FULL\_SEARCH\_FACTOR = 3.0)

### **INTEGRATION\_STEP**

A two-dimensional array (max plot series,max final bins) of 13-character strings (ddd::hh:mm:ss) that specifies the time interval over which interval integrations are to be performed plots for a given plot series. The INTEGRATION\_STEP must be an integer multiple of the SAMPLING\_INTERVAL in ADPIC.NML and less than or equal to the PLOT\_STEP for the same plot series. By default the INTEGRATION\_STEP for a plot series is equal to the PLOT\_STEP.

### **INTEGRATION\_TYPE**

An array (max final bins) of four-character strings ('CUMM', 'INTV', or 'NONE') that specifies the type of integration to perform for a given final bin. 'CUMM' refers to cumulative integration, 'INTV' refers to interval integration, 'NONE' refers to no integration. Only 'CUMM' or 'INTV' can be used for an integrated sampling bin (i.e., SAMPLING\_TYPE='INTAIR' in the adpic\_control namelist).

### **NEST GRIDS**

Flag ('Y' or 'N') which determines whether the nested grid or the coarse sampling grid is to be used.

(Default: NEST\_GRID = 'Y' → used nested grid.)

### **PLOT END DATE**

A two-dimensional array (max plot series, max final bins) of 7-character strings (ddmmmyy) that specifies the date portion of the date/time when a given plot series for a given final bin is to end. The date/times for each final bin should be increasing order. Specifying an end date/time for a plot series is optional. If it is omitted, then a given plot series is terminated by the beginning of a second series or by the of the available concentration data. A plot end date/time must corresponding to the date/time of a CONC\*.BIF file.

### **PLOT END TIME**

A two-dimensional array (max plots eries, max final bins) of 8-character strings (hh:mm:ss) that specifies the time portion of the date/time when a given plot series for a given final bin is to end. The date/times for each final bin should be increasing order. Specifying an end date/time for a plot series is optional. If it is omitted, then a given plot series is terminated by the beginning of a second series or by the of the available concentration data. A plot end date/time must corresponding to the date/time of a CONC\*\*\*.BIF file.

### **PLOT START DATE**

A two-dimensional array (max plot series, max final bins) of 7-character strings (ddmmmyy) that specifies the date portion of the date/time when a given plot series for a given final bin is to begin. The date/times for each final bin should be increasing order. A plot start date/time must corresponding to the date/time of a CONC\*\*\*.BIF file.

### **PLOT START TIME**

A two-dimensional array (max plot series, max final bins) of 8-character strings (hh:mm:ss) that specifies the time portion of the date/time when a given plot series for a given final bin is to begin. The date/times for each final bin should be increasing order. A plot start date/time must corresponding to the date/time of a CONC\*\*\*.BIF file.

### **PLOT STEP**

A two-dimensional array (max plot series, max final bins) of 13-character strings (ddd::hh:mm:ss) that specifies the time interval between plots for a given plot series. The PLOT\_STEP must be an integer multiple of the SAMPLING\_INTERVAL in ADPIC.NML.

### **PRINT CONTOUR CONC**

Flag ('Y', 'N') for writing final bin concentrations to the echo file. This writing also includes the location and value of the max/min concentrations.

(Default: PRINT\_CONTOUR\_CONC = 'N' → none written.)

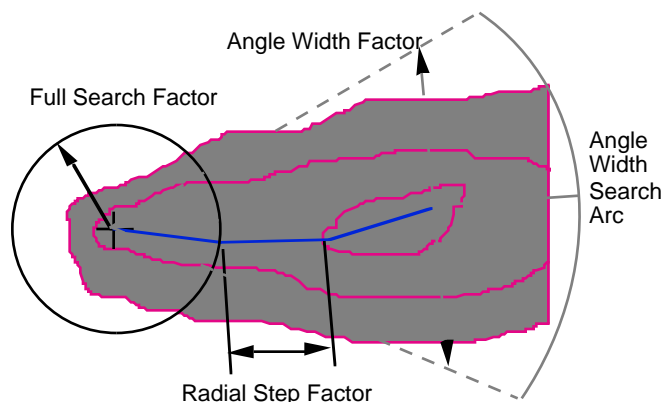
### **PRINT INPUT CONC**

Flag ('Y', 'N') which controls the printing of the input concentrations to the echo file.

(Default: PRINT\_INPUT\_CONC = 'N' → none are written.)

### **RADIAL STEP FACTOR**

Floating point factor applied to the horizontal (x) grid cell size for the centerline capability. This parameter controls the radial step distance for the search for each maximum.



(Default: RADIAL\_STEP\_FACTOR = 0.5)

### **RESUSPEND**

Flag ('Y', 'N') which controls whether a single deposition contour (whose value = 10% of maximum concentration) is computed for representing the resuspension source area, and the RSUSGEO file is created in preparation for a resuspension calculation.

(Default: RESUSPEND = 'N' → neither the resuspension contour is drawn, nor is the RSUSGEO file created.)

### **SAMPLING TO DOSE BIN**

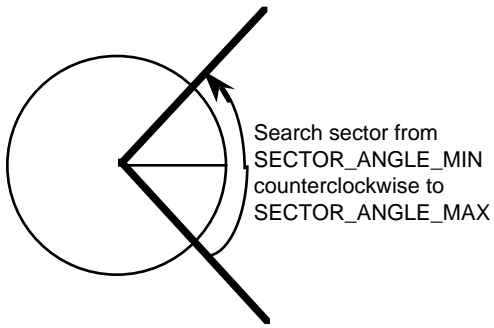
Array (max size = MAXFBINS = 30) of integers, whose values are ADPIC bin numbers, specifying which sample bin goes into which dose bin, i.e., the “ADPIC TO DOSE” bin mapping. For example, if the same sample bin were put into three different dose bins, different dose conversion factors (DCON's) could be applied to each dose bin (for example: whole body dose, child thyroid dose, and air concentration) without having to have separate sample bins. (Also see DOSE\_TO\_FINAL\_BIN.)

(Default: SAMPLING\_TO\_DOSE\_BIN = 1 2 3 4 5 6 7 8 9 10 → one dose bin for each sample bin from ADPIC, with the dose bins in the same order as the sample bins)

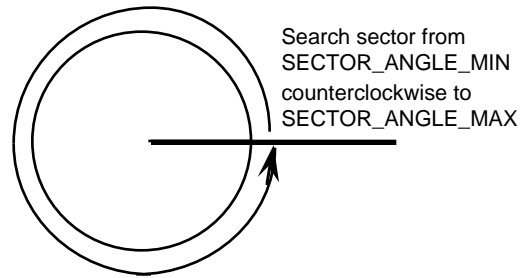
### **SECTOR ANGLE MAX**

Floating point value (0. to 360.) limiting all centerline searches (close-in full search, and further-out angle width search). SECTOR\_ANGLE\_MAX is measured in degrees counterclockwise from east (a mathematical angle, not a meteorological angle). Centerline searches cover angles counterclockwise from SECTOR\_ANGLE\_MIN to SECTOR\_ANGLE\_MAX:





Example with SECTOR\_ANGLE\_MIN = 315 deg  
and SECTOR\_ANGLE\_MAX = 45 deg

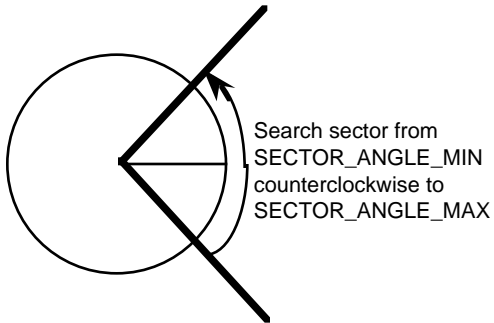


Example with SECTOR\_ANGLE\_MIN = 0 deg  
and SECTOR\_ANGLE\_MAX = 360 deg

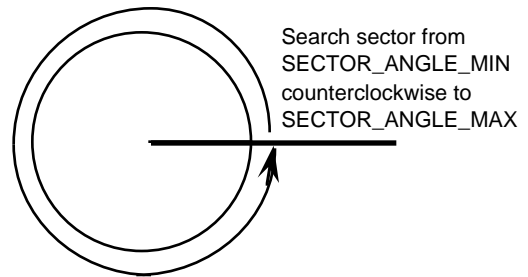
(Default: SECTOR\_ANGLE\_MAX = 360.)

### **SECTOR\_ANGLE\_MIN**

Floating point value (0. to 360.) limiting all centerline searches (close-in full search, and further-out angle width search). SECTOR\_ANGLE\_MIN is measured in degrees counterclockwise from east (a mathematical angle, not a meteorological angle). Centerline searches cover angles counterclockwise from SECTOR\_ANGLE\_MIN to SECTOR\_ANGLE\_MAX:



Example with SECTOR\_ANGLE\_MIN = 315 deg  
and SECTOR\_ANGLE\_MAX = 45 deg



Example with SECTOR\_ANGLE\_MIN = 0 deg  
and SECTOR\_ANGLE\_MAX = 360 deg

(Default: SECTOR\_ANGLE\_MIN = 0.)

### **III.B.11b. PLCNT\_GRAPHICS\_PARAMS namelist**

#### **BLOWUP SOURCE**

An integer specifying the number of the ADPIC source whose location is to be placed as indicated by SOURCE\_OCTANT.

#### **CIRCLE 2000**

Flag ('Y' or 'N') specifying whether or not the 2000ft blast effects circle will be plotted at the release location. (CIRCLE\_2000 = 'Y' is required for DOD plots.)

#### **CONTOUR LEVELS**

A two-dimensional array (5,maxf final bins) of specific contour levels to be plotted for each specified final bin. Thus, up to 5 contour levels can be specified for each final bin.

The CONTOUR\_LEVELS parameter is required for DOD and AWE plots as follows (for final bin n):

CONTOUR_LEVELS(1,n)	=	150.	25.	5.	0.5	(DOD CEWBDE)
CONTOUR_LEVELS(1,n)	=	600.	60.	6.		(DOD deposition)
CONTOUR_LEVELS(1,n)	=	4.7	0.94	0.78	0.15	(AWE 50yr CEWBDE)
CONTOUR_LEVELS(1,n)	=	20.0	2.0	0.2	0.03	(AWE deposition)

Contour levels specified by CONTOUR\_LEVELS that do not meet the MIN\_CONTOUR\_AREA requirement will not be plotted. If none of the levels specified for a final bin meet the MIN\_CONTOUR\_AREA requirement, then PLCNT will pick its own levels. (If CONTOUR\_LEVELS(1,n) is not specified, PLCNT will pick its own contour levels, based upon the MIN\_CONTOUR\_AREA requirement.

#### **HALF POWER OF 10**

Flag indicating whether half powers of 10 (i.e.,  $3 \times 10^n$ ) are to be allowed as possible contour levels.

(Default: HALF\_POWER\_OF\_10 = 'Y')

#### **INITIAL BANNER**

Flag ('Y', 'N') used to indicate an initial calculation. The fact that this is an Initial calculation will be part of the plot description record. (Note: the sending of the description and graphics files is now controlled interactively in Show\_Graphics-Site\_Workstation\_Plot and Show\_Graphics-Site\_System\_Plot.)

(Default: INITIAL\_BANNER = 'N' → not an Initial calculation)

## **MAKE NEW GEOG**

Flag ('Y', 'N') for forcing a new geography file to be produced for overlaying with the contour plot.

(Default: MAKE\_NEW\_GEOG = 'N' → do not create a new geography file if one exists that covers the same area.)

## **MAPSCALE**

A two-element integer array specifying the map scale to be used. The first element represents a unit of actual distance on the map and the second element is the corresponding actual physical distance. For example, a map scale of 1:50,000 means and 1 inch on the map represents 50,000 inches of actual physical distance, and MAPSCALE = 1 50000. This parameter is ignored if MAP\_SCALE\_FLAG = 'N' for all final bins.

## **MAP SCALE FLAG**

Array (max size = MAXFBINS = 30) of flags ('Y', 'N') that indicates whether or not the specified map scale is to be applied to this bin.

(Default: MAP\_SCALE\_FLAG = 'N')

## **MAX CONTOURS**

Array (max size = MAXFBINS = 30) of integers (1,2,3,4, or 5) of contour levels to be plotted and shown in legend.

## **MIN CONTOUR AREA**

Real number representing minimum area ( $\text{km}^2$ ) that will be plotted and shown in the legend. This will eliminate any contour levels specified by CONTOUR\_LEVELS which do not meet the MIN\_CONTOUR\_AREA requirement. (Also see CONTOUR\_LEVELS)

(Default = 0.005% of grid area or  $0.01\text{km}^2$ , whichever is larger)

## **NEST TYPE**

10-character string which specifies if and how the nested grids are shown on the contour plots. Current values and meanings are:

'NONE' → no nesting stuff drawn

'BOUNDARIES' → boundaries of nested grids drawn

'CELLS' → grid cells and boundaries of nested grids drawn

## **NOTES BANNER**

Flag ('Y' or 'N') specifying whether or not the “SEE NOTES” banner is written diagonally across the graphics portion of the plot. (NOTES\_BANNER = 'Y' is required for DOD plots.)

### **NUM\_RECOMPUTED\_CONTOURS**

Integer indicating the number of contours to recompute if they are recomputed (See RECOMPUTE\_USER\_CONTOURS).

(Default: NUM\_RECOMPUTED\_CONTOURS = 5)

### **OUTPUT\_CONTOURS**

Flag ('Y', 'N') indicating whether PLCNT is to produce products to be sent to the Site Systems.

(Default: OUTPUT\_CONTOURS = 'Y' )

### **OUTPUT\_GRIDS**

Flag ('Y', 'N') indicating whether PLCNT is to produce products to be sent to the site workstation. When products send to the site\_workstation are the final bin concentrations (thus, they are the same data that is contoured when site system products are generated with the exception of the smoothing done for the site system) passed as grids of data.

(Default: OUTPUT\_GRIDS = 'Y' )

### **PLOT\_SIZE\_CM**

A real number specifying the size (cm) of the resulting plot as it would appear on the hard copy device. The postscript printers used by ARAC use a value of 16.0 for PLOT\_SIZE\_CM.

### **RECOMPUTE\_USER\_CONTOURS**

Flag ('Y', 'N') indicating whether PLCNT should generate contours levels based on the maximum concentration if the estimated area if the lowest user-specified contour level is less than the MIN\_CONTOUR\_AREA (See NUM\_RECOMPUTED\_CONTOURS).

(Default: RECOMPUTE\_USER\_CONTOURS = 'Y')

### **SMOOTHED\_CONC\_SWS**

Flag ('Y', 'N') indicating whether or not smoothing is to be performed along nest boundaries for site workstation products in the same manner as it is done for the site system products. For testing purposes it is sometimes useful to compare the old and new products. In addition, graphics artifacts due to the nesting can occasionally occur with the contouring algorithm and this option may permit more presentable products to be generated.

(Default: SMOOTHED\_CONC\_SWS = 'N' => different smoothing algorithms for SWS)

### **SOURCE\_OCTANT**

2-character string ('NW','N ','NE','W ','C ','E ','SW','S ','SE') specifying the desired placement of the source within the first subplot when scaling maps (MAP\_SCALE\_FLAG = 'Y'). The following diagram indicates the two character codes corresponding to the allowed locations:

NW	N	NE
W	C	E
SW	S	SE

### **TITLE ON GEOG**

Flag ('Y' or 'N') specifying whether or not the PLOT\_TITLEn string is to appear at the top of the graphics portion (left side) of the plot (TITLE\_ON\_GEOG = 'N' is required for DOD plots.)

### **XMAX WINDOW**

Real number specifying the maximum  $x$  coordinate of the area to be windowed. Used in conjunction with XMIN\_WINDOW, YMIN\_WINDOW, and YMAX\_WINDOW.

(Default: none, i.e., no windowing.)

### **XMIN WINDOW**

Real number specifying the minimum  $x$  coordinate of the area to windowed. Used in conjunction with XMAX\_WINDOW, YMIN\_WINDOW, and YMAX\_WINDOW.

(Default: none, i.e., no windowing.)

### **XY TICK LABEL**

Flag ('Y' or 'N') specifying whether or not XY tick marks and labels are to appear on the bottom and left sides of the plot at the time of screen display on the Central System and on hard copies produced by the Central System; i.e., the tick marks are not part of the graphics file, but are generated separately by graphics display application and are never sent to a Site System or Site Workstation. This parameter is only used by PLCNT to pass on to the graphics display application. (XY\_TICK\_LABEL = 'N' is the usual value.)

### **YMAX WINDOW**

Real number specifying the maximum  $y$  coordinate of the area to be windowed. Used in conjunction with XMIN\_WINDOW, XMAX\_WINDOW, and YMIN\_WINDOW.

(Default: none, i.e., no windowing.)

### **YMIN WINDOW**

Real number specifying the minimum  $y$  coordinate of the area to be windowed. Used in conjunction with XMIN\_WINDOW, XMAX\_WINDOW, and YMAX\_WINDOW.

(Default: none, i.e., no windowing.)

### **III.B.11c. PLCNT\_LABELING\_PARAMS namelist**

#### **DESCR FORMAT**

Array (max size = MAXFBINS = 30) of 5-character strings specifying the format to be used in displaying the cross-hatching and area information in the plot legend. Current values and meanings are: '3LINE' → use 3-line DOD action level format, 'CODED' → use 2-character coded format which does not include the isopleth value (used for AWE), and 'NONE' → no coding or action level descriptions.

#### **DESCR HEADERA**

Array (max size = MAXFBINS = 30) of 32-character strings specifying the first line of a two-line header that appears immediately above the cross-hatching section of the legend. (Usual values: DESCR\_HEADERA = 'Exposure Action Levels:' for DOD sites and DESCR\_HEADERA = 'Contours:' for other sites.

#### **DESCR HEADERB**

Array (max size = MAXFBINS = 30) of 32-character strings specifying the second line of a two-line header that appears immediately above the cross-hatching section of the legend. (Usual value: DESCR\_HEADERB = '(Level and Area Covered) ' .

#### **GENERATION DATE**

7-character string (ddmmmyy) labeled as the generation date of a set of plots in the second or third line of the plot legend. Note that this time is not currently the same as the generation time in the product description record.

(Default: GENERATION\_DATE = 'unset' which uses the current system time.)

#### **GENERATION TIME**

8-character string (hh:mm:ss) labeled as the generation time of a set of plots in the second or third line of the plot legend. Note that this time is not currently the same as the generation time in the product description record.

(Default: GENERATION\_TIME = 'unset' which uses the current system time.)

## **LEGEND\_TITLE**

Array (max size = MAXFBINS = 30) of 32-character strings, which appear in the margin above the legend section.

(Default: LEGEND\_TITLE = ' '.)

## **LEGEND\_TITLE\_ALL**

32-character string for all final bins, which appears in the margin above the legend section.

(Default: LEGEND\_TITLE\_ALL = ' '.)

## **LEVEL\_DESCRit**

Array (max size = MAXFBINS = 30) of 32-character strings describing emergency or protective actions associated with a given contour level (specified by CONTOUR\_LEVELS). There are a maximum of three (t = A, B and C) 32-character strings each for each of 5 contour levels (i = 1 to 5) for each of 30 final bins. The maximum combination of character strings may be used for DESCR\_FORMAT = '3LINE', while only the first eight characters of the LEVEL\_DESCRiA strings are used for DESCR\_FORMAT = 'CODED'. LEVEL\_DESCRit is not used for DESCR\_FORMAT(n) = 'NONE'. Below is a complete specification (i.e., explicitly specifying all three lines for each of the five possible contour levels) for DESCR\_FORMAT = '3LINE' for final bin 1:

LEVEL_DESCR1A (1)	=	'	Immediate action required:	'
LEVEL_DESCR1B (1)	=	'	Respiratory protection required.	'
LEVEL_DESCR1C (1)	=	'	Recommend sheltering or removal.	'
LEVEL_DESCR2A (1)	=	'	Prompt respiratory protection	'
LEVEL_DESCR2B (1)	=	'	required and recommend	'
LEVEL_DESCR2C (1)	=	'	evacuation.	'
LEVEL_DESCR3A (1)	=	'	Respiratory protection required,	'
LEVEL_DESCR3B (1)	=	'	recommend sheltering,	'
LEVEL_DESCR3C (1)	=	'	consider evacuation.	'
LEVEL_DESCR4A (1)	=	'	Consider sheltering.	'
LEVEL_DESCR4B (1)	=	'		'
LEVEL_DESCR4C (1)	=	'		'
LEVEL_DESCR5A (1)	=	'		'
LEVEL_DESCR5B (1)	=	'		'
LEVEL_DESCR5C (1)	=	'		'

## **MATERIAL**

Array (max size = MAXFBINS = 30) of 22-character strings specifying the material or species for a specified final bin that is to appear in the plot legend.

### **PLOT\_DESCR**

Array (max size = MAXFBINS = 30) of 32-character strings which will appear in the “one-line plot description” in the scrolling regions used to select plots in the graphical display application.

### **PLOT\_TITLE**

Array (max size = MAXFBINS = 30) of 32-character strings which will appear in the margin above the plot. This string will also appear across the top of the graphics section if TITLE\_ON\_GEOG = 'Y'.

(Defaults: PLOT\_TITLE = PLOT\_DESCR)

### **PROBLEM\_TITLE PLCNT**

70-character string that will appear in the title field in the expanded plot description on the Site System and is written as the first line of the PLCNT.LOG file.

### **REMARKSt**

Array (max size = MAXFBINS = 30) of 32-character strings for up to 4 lines (t = A to D) of remarks. The first line (t = A) usually begins with the characters 'REMARKS:'

### **REMARKSt ALL**

32-character strings for up to 4 lines (t = A to D) of remarks that are not final bin-specific. The first line (t = A) usually begins with the characters 'REMARKS:'

### **SITESYS\_LOCATION\_NUMBER**

Integer (0 to number of sources) representing the ADPIC source number to identify the default location on the site system.

(Default: SITESYS\_LOCATION\_NUMBER = 1 )

### **SOURCE\_LOCATION\_NUM**

Integer (0 to number of sources) representing the ADPIC source number for which the location of the source should be displayed in the plot legend. For example, “SOURCE\_LOCATION\_NUM = 1” will result in location of ADPIC source number 1 being written in legend. (Also see SOURCE\_LOCATION\_TYPE.)

(Default: SOURCE\_LOCATION\_NUM = 0 → no location is written in the legend.)

### **SOURCE\_LOCATION\_TYPE**

6-character string ('XY ' or 'LATLON') indicating whether the source location is to be displayed as UTM's or latitude/longitude. (Also see SOURCE\_LOCATION\_NUM). This does not apply to Site Workstation plots, which have room to display both x-y coordinates and lat-long coordinates.

(Default: SOURCE\_LOCATION\_TYPE = 'LATLON' → latitude/longitude location)



## **UNITS**

Array (max size = MAXFBINS = 30) of 10-character strings specifying the units of the concentrations of the specified final bin that appear on the plots.

NOTE: When DESCR\_FORMAT = '3 LINE,' only the first 6 characters of UNITS will appear on the plots.

(Default: UNITS = ' '.)

### **III.B.11d. TIMEHIS\_PARAMS namelist**

#### **BACKGROUND CALC**

Array (max size = MAXFBINS = 30) of floating point values representing the background concentration to be added to the model-calculated concentration values before statistics are calculated. It is final bin-specific, so a different value can be used for all samplers associated with a given final bin.

#### **BACKGROUND CALC ALL**

Floating point value for all final bins representing the background concentration to be added to the model-calculated concentration values before statistics are calculated.

#### **BACKGROUND MEAS**

Array (max size = MAXFBINS = 30) of floating point values representing the background concentration to be added to the measured concentration values (input via the SAMINFO parameter) before statistics are calculated. It is final bin-specific, so a different value can be used for all samplers associated with a given final bin.

#### **BACKGROUND MEAS ALL**

Floating point value for all final bins representing the background concentration to be added to the measured concentration values (input via the SAMINFO parameter) before statistics are calculated.

#### **BIN PLOT MAX**

Array (max size = MAXFBINS = 30) of floating point values that sets the maximum on the vertical axis for the time series plots. It is final bin-specific, so a different maximum can be used for all samplers associated with a given final bin.

#### **BIN PLOT MAX ALL**

The value sets the maximum on the vertical axis for the time series plots for all final bins.

#### **BIN PLOT MIN**

Array (max size = MAXFBINS = 30) of floating point values that sets the minimum on the vertical axis for the time series plots. It is final bin-specific, so a different minimum can be used for all samplers associated with a given final bin.

#### **BIN PLOT MIN ALL**

The value sets the minimum on the vertical axis for the time series plots for all final bins.

### **COMPARE MEAS CALC**

Flag ('Y', 'N') controlling whether statistics between measured and calculated values are calculated. If this flag is set in 'Y', measured-calculated comparison statistics including percentage of values within factor "R" will be written to the STATS.DAT file, and the minimum and maximum computed concentrations for various hit angles will be written to the SAMPOUT.DAT file. The SAMPOUT.DAT file is used as input for another statistics code, FN. The COMPARE\_MEAS\_CALC flag also controls plotting of measured data on time history graphics plots.

### **FACTOR R**

Array (max size = MAXFACTORS = 15) of floating point values representing factors to be used in factor R analysis. Current values are (1.5, 2.0, 3.0, 5.0, 7.0, 10.0, 20.0, 50.0, 100.0, 1000.0).

### **HIT ANGLE**

Array (max size = MAXHANGLES = 10) of floating point values representing angles to be used in hit angle analysis. These angles are used to define an area centered on an interpolation point where sample of additional interpolations are calculated. If a measured value lies within the range defined by the minimum and maximum concentration found in the sample the a hit (or exact match) is recorded for this angle (See Section II.C.4 for a description of this the hit angle area). Current values are (0.0, 2.0 5.0, 10.0, 15.0, 20.0). Note that the first hit angle is always zero and is used to record the interpolated value for the specified location.

### **MIN CONC**

Array (max size = MAXFBINS = 30) of floating point value representing the minimum measured or computed concentration to be used for statistics calculations. If either the measured or computed concentrations (with background values already added in) are greater than MIN\_CONC, then that measured-computed pair is included in the statistics. The default is 0.0. It is final bin-specific, so a different value can be used for all samplers associated with a given final bin.

### **MIN CONC ALL**

Floating point value for all bins representing the minimum measured or computed concentration to be used for statistics calculations. If either the measured or computed concentrations (with background values already added in) are greater than MIN\_CONC, then that measured-computed pair is included in the statistics. The default is 0.0.

### **PROBLEM TITLE TIMEHIS**

70-character string that currently appears in the title field in the expanded plot description on the Site System and is written as the first line of the TIMEHIS.LOG file.

### **SHOW TIME ORIGIN**

Flag ('Y', 'N') used to ensure that the time origin (i.e, 0 hours) is included on time history plots. Used to produced plots that are appropriate for publication. Default is 'N'.

### **STATISTICS SOURCE**

Integer indicating the source to be used in hit angle analysis. That is, the angles and radii used are with reference to a particular ADPIC source, so this source must be specified. Default is 1.

### **TIMESTEPS PER PLOT**

Integer number (maximum of 200) of data points (corresponding to SAMPLING\_INTERVAL restart files) to be plotted on each frame of the time history plots.

### **XMAX SAMPLES**

Real number specifying the maximum  $x$  coordinate of the area to be analyzed. Used in conjunction with XMIN\_SAMPLES, YMIN\_SAMPLES, and YMAX\_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

### **XMIN SAMPLES**

Real number specifying the minimum  $x$  coordinate of the area to analyzed. Used in conjunction with XMAX\_SAMPLES, YMIN\_SAMPLES, and YMAX\_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

### **YMAX SAMPLES**

Real number specifying the maximum  $y$  coordinate of the area to be analyzed. Used in conjunction with XMIN\_SAMPLES, XMAX\_SAMPLES, and YMIN\_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

### **YMIN SAMPLES**

Real number specifying the minimum  $y$  coordinate of the area to be analyzed. Used in conjunction with XMIN\_SAMPLES, XMAX\_SAMPLES, and YMAX\_SAMPLES.

(Default: none, i.e., the ADPIC grid.)

## Sample PLCNT Input File

### \$sconc\_analysis\_control

SAMPLING_TO_DOSE_BIN	=	1 2 3 4 5 6 7 8 9 10	←	relation of ADPIC bins to dose bins
DCON	=	1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0	←	dose conv. factors for each dose bin
DOSE_TO_FINAL_BIN(1,1)	=	1            4            7	}	relation of dose bins to final bins
DOSE_TO_FINAL_BIN(1,2)	=	2            5            8		
DOSE_TO_FINAL_BIN(1,3)	=	3            6            9		
DOSE_TO_FINAL_BIN(1,4)	=	10		
PLOT_START_DATE(1,1)	=	'01APR85'	}	date (ddmmmyy) to start a plot series
PLOT_START_DATE(1,2)	=	'01APR85'		
PLOT_START_DATE(1,3)	=	'01APR85'		
PLOT_START_DATE(1,4)	=	'01APR85'		
PLOT_START_TIME(1,1)	=	'01:00'	}	time (hh:mm) to start a plot series
PLOT_START_TIME(1,2)	=	'01:00'		
PLOT_START_TIME(1,3)	=	'01:00'		
PLOT_START_TIME(1,4)	=	'01:00'		
PLOT_STEP(1,1)	=	'00::01:00'	}	time interval (dd::hh:mm) between plots
PLOT_STEP(1,2)	=	'00::01:00'		
PLOT_STEP(1,3)	=	'00::01:00'		
PLOT_STEP(1,4)	=	'00::01:00'		
INTEGRATION_TYPE(1)	=	'cumm'	}	specify type of time integration
INTEGRATION_TYPE(2)	=	'none'		
INTEGRATION_TYPE(3)	=	'none'		
INTEGRATION_TYPE(4)	=	'none'		
NEST_GRIDS	=	'Y'	←	flag for use of nested grid sampling

\$end

### \$plcnt\_graphics\_params

INITIAL_BANNER	=	'N'	←	Initial plot flag for expanded descr.
NOTES_BANNER	=	'N'	←	flag for "SEE NOTES" banner
MAKE_NEW_GEOG	=	'N'	←	flag for creating new geography file
TITLE_ON_GEOG	=	'N'	←	flag for title on left top portion
CIRCLE_2000	=	'N'	←	flag for 2000ft circle
XY_TICK_LABEL	=	'Y'	←	flag for x-y axis labels
MIN_CONTOUR_AREA	=	0.0	←	minimum contour area
NEST_TYPE	=	'NONE'	←	nesting grid to be overlayed on plot
MAX_CONTOURS(1)	=	5	}	maximum number of isopleths (for each final bin)
MAX_CONTOURS(2)	=	5		
MAX_CONTOURS(3)	=	5		
MAX_CONTOURS(4)	=	5		
CONTOUR_LEVELS(1,1)	=	1E0 1E-1 1E-2 1E-3 1E-4	}	isopleths to be plotted (for each final bin)
CONTOUR_LEVELS(1,2)	=	1E-4 1E-6 1E-8 1E-10		
CONTOUR_LEVELS(1,3)	=	1E-1 1E-2 1E-3 1E-4 1E-5		
CONTOUR_LEVELS(1,4)	=	1E0 1E-1 1E-2 1E-3 1E-4		

\$end

### \$plcnt\_labeling\_params

PROBLEM_TITLE_PLCNT	=	'ARAC Contour Plots'	←	title for expanded desc.
---------------------	---	----------------------	---	--------------------------

MATERIAL(1)	=	'Gnd, Elevated, Column '	}	material labels for plot legend (for each final bin)
MATERIAL(2)	=	'Gnd, Elevated, Column '		
MATERIAL(3)	=	'Gnd, Elevated, Column '		
MATERIAL(4)	=	'Gnd, Elevated, Column '		
UNITS(1)	=	'Ci*s/m3'	}	units labels for plot legend (for each final bin)
UNITS(2)	=	'Ci/m3'		
UNITS(3)	=	'Ci/m2'		
UNITS(4)	=	'Ci*s/m3'		
GENERATION_DATE	=	'unset'	←	plot generation date & time for plot legend
GENERATION_TIME	=	'unset'	←	source location format
SOURCE_LOCATION_TYPE	=	'XY'	←	src number for nesting
SOURCE_LOCATION_NUM	=	1	←	
PLOT_TITLE(1)	=	'Jan 1991 Benchmark Runs'	}	2nd line of plot legend (for each final bin)
PLOT_TITLE(2)	=	'Jan 1991 Benchmark Runs'		
PLOT_TITLE(3)	=	'Jan 1991 Benchmark Runs'		
PLOT_TITLE(4)	=	'Jan 1991 Benchmark Runs'		
PLOT_DESCR(1)	=	'Run1G: Integ. Conc. 1.5m '	}	appears in one-line descr. when selecting a plot to display/delete/etc. (for each final bin)
PLOT_DESCR(2)	=	'Run1G: Inst. Conc. 1.5m '		
PLOT_DESCR(3)	=	'Run1G: Deposition '		
PLOT_DESCR(4)	=	'Run1G: Integ. Conc. 75m '		
LEGEND_TITLE(1)	=	'Run1G: Src hgts, Stabilities '	}	1st line of plot legend (for each final bin)
LEGEND_TITLE(2)	=	'Run1G: Src hgts, Stabilities '		
LEGEND_TITLE(3)	=	'Run1G: Src hgts, Stabilities '		
LEGEND_TITLE(4)	=	'Run1G: Src hgts, Stabilities '		
REMARKSA_ALL	=	Remarks: STAB_CLASS=4, 2, 4, 5 '	}	4 lines of free-form remarks for plot legend (optionally final bin-specific)
REMARKSB_ALL	=	'Winds: calm, lgt, med, stg '		
REMARKSC_ALL	=	5m diam. gnd & elevated spheres, '		
REMARKSD_ALL	=	200m column sources '		
DESCR_HEADERA(1)	=	'Exposure Action Levels: '	}	specifies content of 2 lines forming the subtitle for the isopleth labeling section of the plot legend
DESCR_HEADERB(1)	=	'(Level and Area Covered) '		
DESCR_HEADERA(2)	=	'Exposure Action Levels: '		
DESCR_HEADERB(2)	=	'(Level and Area Covered) '		
DESCR_HEADERA(3)	=	'Exposure Action Levels: '		
DESCR_HEADERB(3)	=	'(Level and Area Covered) '		
DESCR_HEADERA(4)	=	'Exposure Action Levels: '		
DESCR_HEADERB(4)	=	'(Level and Area Covered) '		
DESCR_FORMAT(1)	=	'NONE '	}	specifies format of isopleth labeling in plot legend (for each final bin)
DESCR_FORMAT(2)	=	'NONE '		
DESCR_FORMAT(3)	=	'NONE '		
DESCR_FORMAT(4)	=	'NONE '		

Send

### \$timehis\_params

PROBLEM_TITLE_TIMEHIS	=	'ARAC Time History Plots'	←	title for plots
TIMESTEPS_PER_PLOT	=	8	←	timesteps/points per plot
COMPARE_MEAS_CALC	=	'Y'	←	compare meas. to calc. values
BACKGROUND_CALC_ALL	=	0.000E+00	←	backgnd values to be added
BACKGROUND_MEAS_ALL	=	0.000E+00		
MIN_CONC_ALL	=	0.000E+00	←	min. value for statistics calc.

Send

### III.B.12. TIMEHIS

The CONC\_ANALYSIS.NML file used to control PLCNT is also used to control TIMEHIS. TIMEHIS uses the CONC\_ANALYSIS\_CONTROL and the TIMEHIS\_PARAMS namelist blocks in this file, which are described in sections III.B.11.a and III.B.11.d, respectively. Sampler data is entered from a separate file, described next.

The SAMPLER.DATA file has one required section and one optional section. The required section contains the sampler names, coordinates and associated final bins at which concentration values are to be interpolated from the model concentration data. If measurement data is available, this is included in an optional section. Each entry in this section contains the sampler name, the associated final bin number, the observation data, the observation time and the value. Note that if the sampler name and final bin fields are replaced with commas the value from the previous record is inherited. Thus, redundantly specifying these values is unnecessary. Also note that the keyword SAMPLERS is required (unquoted, on a separate line) in all SAMPLER.DATA files and the MEASUREMENTS keyword is required (also unquoted, on a separate line) if measurements data is to be included.

#### Example SAMPLER.DATA file

SAMPLERS						sampler name, x-coord, y-coord, array (max size=5) of final bins (Maximum of 600 sampler locations)
'S0001'	500.0	4201.1	2	3	4	
'S0002'	503.5	4215.0	4			
'S0003'	495.11	4191.2	2	3		
'S0004'	513.8	4203.9	2	4		
MEASUREMENTS						sampler name, final bin, meas. date, meas. time, measurement value (Maximum of 200 measurement times)
'S0002'	2	'11JAN93'	'1200'	121.0		
,	,	'11JAN93'	'1300'	105.0		
,	,	'11JAN93'	'1400'	95.0		
,	,	'11JAN93'	'1500'	78.0		
,	,	'11JAN93'	'1600'	51.0		
,	3	'11JAN93'	'1200'	149.0		
,	,	'11JAN93'	'1300'	137.0		
,	,	'11JAN93'	'1400'	97.0		
'S0003'	4	'11JAN93'	'1300'	110.0		
,	,	'11JAN93'	'1400'	93.0		
,	,	'11JAN93'	'1500'	70.0		
,	,	'11JAN93'	'1700'	56.0		
,	,	'11JAN93'	'1900'	45.0		
,	,	'11JAN93'	'2100'	35.0		
,	,	'11JAN93'	2200'	159.0		

### **III.B.13. RSUSIN**

RSUSIN is an input file to RSUS containing the user-specified parameters needed to determine the resuspension source term and to construct the input files (STNLOC.MET, TOPOG.NML, MEDIC, ADPIC, and PLCNT) for the resuspension CG-MATHEW/ADPIC calculation.

IMPORTANT: All lines in RSUSIN must begin in column 2. Also, the “\$CML” line must be the first line, and “\$END” must be the last line of the file.

The following variables must be assigned a value in RSUSIN:

#### **\$CML**

This signifies the beginning of a VAX namelist.

#### **CD**

Floating point value representing the drag coefficient. More information about the drag coefficient can be found in Priestly (1954).

(Default: CD = 0.0028)

#### **DELTX**

Floating point value representing the horizontal ADPIC grid cell size (km) to appear in TOPOG.NML.

(Default: DELTX = 0.75  $\Rightarrow$  30km grid)

#### **DELZ**

Floating point value representing the vertical ADPIC grid cell size (m) to appear in TOPOG.NML.

(Default: DELZ = 50.0  $\Rightarrow$  700m grid depth)

#### **DIR**

Floating point value representing the surface wind direction (deg) at the resuspension source site.

#### **IFLATTOPO**

Integer flag (1 or 0) indicating whether or not flat terrain will be used.

(Default: IFLATTOPO = 1  $\Rightarrow$  flat terrain)

#### **\$END**

End of namelist indicator



### **III.B.13. RSUSIN**

RSUSIN is an input file to RSUS containing the user-specified parameters needed to determine the resuspension source term and to construct the input files (STNLOC.MET, TOPOG.NML, MEDIC, ADPIC, and PLCNT) for the resuspension CG-MATHEW/ADPIC calculation.

IMPORTANT: All lines in RSUSIN must begin in column 2. Also, the “\$CML” line must be the first line, and “\$END” must be the last line of the file.

The following variables must be assigned a value in RSUSIN:

#### **\$CML**

This signifies the beginning of a VAX namelist.

#### **CD**

Floating point value representing the drag coefficient. More information about the drag coefficient can be found in Priestly (1954).

(Default: CD = 0.0028)

#### **DELTX**

Floating point value representing the horizontal ADPIC grid cell size (km) to appear in TOPOG.NML.

(Default: DELTX = 0.75  $\Rightarrow$  30km grid)

#### **DELZ**

Floating point value representing the vertical ADPIC grid cell size (m) to appear in TOPOG.NML.

(Default: DELZ = 50.0  $\Rightarrow$  700m grid depth)

#### **DIR**

Floating point value representing the surface wind direction (deg) at the resuspension source site.

#### **IFLATTOPO**

Integer flag (1 or 0) indicating whether or not flat terrain will be used.

(Default: IFLATTOPO = 1  $\Rightarrow$  flat terrain)

#### **\$END**

End of namelist indicator

The following parameters have default values, and need only be specified if you do not wish to use the defaults:

### **ISTRT**

10-character string (yymmddhhmm) representing the starting time for the resuspension calculation, and will appear in MEDIC, ADPIC, and PLCNT.

### **SPD**

Floating point value representing the surface wind speed (m/s) at the resuspension source site.

### **NSURF**

Integer (1,2,or 3) representing the soil surface type within the resuspension source term area: 1 ⇒ sandy, 2 ⇒ loam, 3 ⇒ clay or snow.

(Default: NSURF = 1)

### **TITLE**

50-character string which will be used for the Title parameter in the ADPIC.NML and PLCNT namelist files created by RSUS.

## **Sample RSUSIN Input File**

\$CML		←	beginning of namelist indicator
TITLE	= 'SAMPLE TITLE'	←	title to appear on particle and contour plots
NSURF	= 1	←	surface type (1=sandy,2=loam,3=clay or snow)
CD	= 0.0028	←	drag coefficient
SPD	= 15.0	←	sfc wind speed (m/s) at resus. src site
DIR	= 185.0	←	sfc wind direction (deg) at resus. src site
ISTRT	= '9103151830'	←	starting yymmddhhmm for resuspension run
DELTX	= 0.75	←	horizontal ADPIC grid cell size (km)
DELZ	= 50.0	←	vertical ADPIC grid cell size (m)
IFLATTP	= 0	←	flag for using flat terrain
I		}	end of namelist indicator
\$END	= 0.		

### III.C. Output Files

#### III.C.1. TOPOG\*.GRID

The GRID file is generated by TOPOG and supplies the model grid definitions and model terrain for all subsequent codes in the ARAC code stream. MEDIC, CG-MATHEW, ADPIC, PLCNT and TIMEHIS read the GRID file. GRID is an ASCII file used to pass data between codes and provides the coordinate information for the entire model sequence. For any meaningful use of the binary files that used in other parts of the ARAC model system, the TBB pointer to the appropriate GRID file must be maintained as the final description of the grid exists only in this file. The GRID file is composed of two sections:

Variable	Dimension	Type	Description
Traceback Block			Traceback file names
MODELTOPO	IMAXMAT,JMAXMAT	INTEGER	Model topography ; CG-MATHEW grid 0 origin vertical cell index

#### MODEL TOPO

The model topography is represented by an integer array with each array element (i,j) corresponding to a horizontal CG-MATHEW grid cell. The value of each array element as the number of  $k$  levels above the grid bottom, i.e., the bottom of the grid indicated by zero (0): this is the ADPIC convention for vertical cell indexing. The array is dimensioned IMAXMAT by JMAXMAT which, for the typical CG-MATHEW grid, is 51 by 51. With this data, MEDIC and CG-MATHEW must increment the cell heights by one after reading this array and ADPIC must selection the appropriate subset of the array as a function of the ADPIC grid dimensions relative to the CG-MATHEW grid dimensions.

#### III.C.2. TOPOG.LOG

TOPOG is the ASCII output file for TOPOG. It is used to examine and verify the results of, or to debug, a TOPOG run. It is not used by other computer codes. During normal execution of the ARAC code stream, this file would rarely be examined, but in the event of difficulty, the information is available. There are four sections in an TOPOG file:

1. TOPOG.NML echo
2. Grid parameter display
3. Cell heights

#### 4. Model levels

Each of these sections is discussed below.

##### Cell Heights

The averaged cell height in meters for each CG-MATHEW cell is printed. This is stored in a real array dimensioned IMAXMAT by JMAXMAT.

##### Grid Parameter Display

The various grid parameters are written exactly as they are displayed on the user's terminal during a TOPOG run. These parameters include the minimum and maximum extent of the terrain data base in UTM coordinates, the CG-MATHEW and ADPIC model grid dimensions, the grid cell sizes, and the sampling cell description. If errors are found in defining the model grids, then the error messages are printed before the parameter display as they are on the user's terminal.

##### Model Levels

The CG-MATHEW model level (k-levels) for each cell is printed. This is stored in an integer array dimensioned IMAXMAT by JMAXMAT.

### **III.C.3. TOPOG\*.CNML**

The entire TOPOG.NML namelist block is written to a uniquely named exactly as it appears in the input file if the TOPOG.NML data are read correctly. Otherwise the last line will contain the error.

### **III.C.4. TOPOG\*.CGMB**

TOPOG\*.CGMB is the graphics output file for TOPOG. It is composed of two plots:

1. Topography contours,
2. CG-MATHEW cell heights. These are described as follows:

##### CG-MATHEW Cell Heights

The CG-MATHEW cell height along each column of grid cells is the number of model levels below the terrain with one (1) indicating the lowest terrain level in the CG-MATHEW grid.

##### TOPOGRAPHY Contours

The topography contours are isolines of model terrain height. The CG-MATHEW block topography is converted to heights in meters above sea level at each of the horizontal grid points. The height at a model grid point is equal to the highest of the grid cells (usually four) adjacent to it. The resulting array of values is contoured and the contour lines are labeled in meters above sea

level. The default contour interval is 1\*DELZ but this can be overridden by giving a value to CONTOUR\_INTERVAL in TOPOG.NML (see TOPOG.NML description for details). The contours are displayed on the CG-MATHEW grid; the ADPIC grid is indicated by a square drawn within the grid. To avoid confusion with the contour lines, no line elements of geography are drawn, but the point elements (location names, character data) are displayed.

### III.C.5. MEDVEL\*\*\*.BIF

The MEDVEL files are generated by MEDIC and are used to pass the extrapolated or interpolated wind field to CG-MATHEW. One MEDVEL file is produced for each data set specified by a MEDIC\_PARAMS namelist. The MEDVEL is a binary file composed of two sections described below:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
U0, V0	IMAXMAT,JMAXMAT,KMAXMAT	REAL*4	Extrapolated winds

#### Traceback Block

See Appendix C

#### U0, V0

The extrapolated or interpolated winds from MEDIC or HMEDIC are written in separate *u*- and *v*-component arrays as floating point numbers in meters/second. Wind vectors exist at each grid point in the CG-MATHEW grid volume. Therefore, each array is dimensioned IMAXMAT by JMAXMAT by KMAXMAT. For the typical CG-MATHEW grid the arrays are 51 by 51 by 15 or 47 by 51 by 15 for the hemispheric case. The grid volume includes terrain. The component arrays follow each other (*v* after *u*) in the MEDVEL file.

### III.C.6. MEDIC.LOG

MEDIC.LOG is the ASCII output file from MEDIC. One MEDIC.LOG file is produced for each MEDIC run containing This file is used to examine and verify the results of, or to debug, a MEDIC run. It is not used by other computer codes. During normal execution of the ARAC code stream, this file would rarely be examined, but if problems arise the information is available as an aid in locating the difficulty.

There are five possible sections in each MEDIC.LOG file which are discussed below:

1. MEDIC.NML listing
2. Grid descriptions
3. Tables of valid GWC data if input

4. Reference level winds (MEDIC) or GWC winds (HMEDIC)
5. Extrapolated or interpolated MEDIC winds

#### MEDIC.NML listing

The completed MEDIC.NML data set is written out including default settings. The MEDIC.NML listing will appear in all MEDIC.LOG files.

#### Grid Descriptions

The model grid parameters are listed under a 'TOPOG.NML:' heading. This includes values describing the terrain data base, ADPIC, and CG-MATHEW grids.

#### Tables of valid data input from GWC metdata files

Tables are printed indicating the valid input data read from each AFGWC metdata file accessed.

#### Reference Level or GWC Winds

The reference level wind vectors (MEDIC) or the input GWC winds (HMEDIC) are printed as separate *u*- and *v*-component arrays of real numbers in meters second. A reference level wind vector exists at each grid point in the horizontal direction, so the arrays are dimensioned IMAXMAT by JMAXMAT. The *u*-component array is printed first, followed by the *v*-component array. This section will appear only if PRINT\_WINDFIELDS is set to 'Y' in the STNLOC.MET file.

#### Extrapolated or InterpolatedWinds

The extrapolated or interpolated wind vectors are printed as separate *u*- and *v*-component arrays of real numbers in meters/second. These arrays are printed as a series of two-dimensional arrays, one for each vertical model level. Thus for a typical UTM grid, 15 arrays, each dimensioned 51 by 51, will be printed to define one component. The *u*-component arrays are written first, followed by the *v*-component arrays. This section will appear only if PRINT\_WINDFIELDS is set to one 'Y' in the STNLOC.MET file.

### **III.C.7. MEDIC\*.CGMB**

MEDIC\*.CGMB is the graphics output file for MEDIC. It is composed of seven types of plots:

1. Surface data wind barb plots (MEDIC only)
2. Upper air wind barb plots (MEDIC and HMEDIC)
3. Horizontal plane vector plots (MEDIC and HMEDIC)
4. Sample profile locations (MEDIC and HMEDIC)
5. Sample profiles (MEDIC and HMEDIC)
6. Measured profiles (MEDIC only)
7. Input data (MEDIC only)

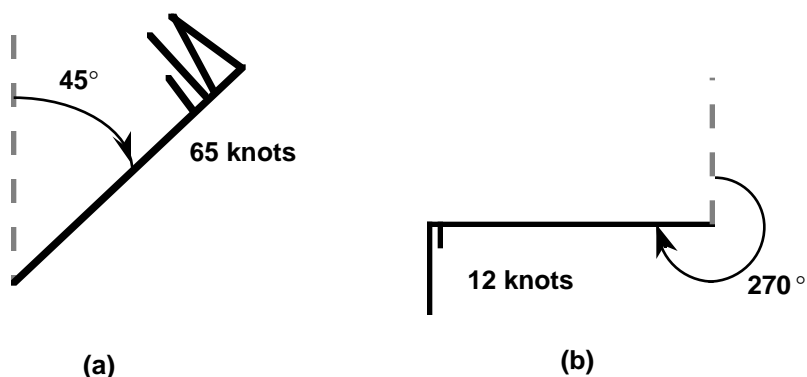
These plots are described below:

## Surface Data

The surface wind observations, are plotted on the CG-MATHEW grid border as wind barbs. For regional MEDIC, the winds are adjusted to the reference height (REF\_HGT) according to the surface layer power law using the exponent PWRSL.

Wind barbs, as used on ARAC graphics products, present a slight variation on the standard meteorological wind barb convention for representing wind observations. In the standard convention, wind direction is represented by a line segment, or stem, with one end, the origin, located at the observation point and the other end located exactly upwind from the origin. Thus the stem represents the direction from which the wind is blowing. Speed ranges are represented by the number of flags and short line segments, i.e., barbs, attached to the stem. The first flag or barb is attached to the stem at the distal end with additional flags and barbs being placed closer to the origin. Flags are added first, at the distal end, followed by barbs. Each flag represents a wind speed of 50 knots and each barb represents 10 knots. The last barb, i.e., closest to the origin, may be half the length of a normal barb, thereby indicating a 5 knot wind. Thus a stem with one flag, one barb, and one half barb indicates a wind of at least  $50 + 10 + 5 = 65$  knots, but less than 70 knots (see Fig. B.1a.). With ARAC's convention, because many of the surface wind speeds are low, the 5 knot ( $\sim 2.5$  m/s) resolution of this scheme is limiting. Therefore the last barb is scaled exactly to the reported speed, e.g., a 12 knot wind would be represented as one barb at the end of the stem (10 knots) and one short barb one-fifth as long as a standard barb somewhat closer to the origin (see Fig. III.C.1b.).

## Horizontal Plane Vector Plots



**Fig. III.C.1. Examples of wind barbs showing (a) 65 knot winds from 45 N and (b) 12 knot winds from 270 N.**

In plotting the surface observations, the station name is printed adjacent to the origin of the barb. If the speed data is missing from an observation, only the stem is plotted, indicating the direction. If the direction data is missing from an observation, a square is plotted, containing an integer giving the wind speed in m/s. A zero wind speed observation is indicated by a double circle. The density of the barbs can be controlled by assigning value to the appropriate variables

GWC\_BARB\_PLOT\_FREQ, RWM\_BARB\_PLOT\_FREQ, or SFC\_BARB\_PLOT\_FREQ. Similarly, the vectors may be scaled using GWC\_BARB\_SCALE, RWM\_BARB\_SCALE, or SFC\_BARB\_SCALE.

Horizontal grids of extrapolated wind vectors are plotted at constant heights above terrain. Since wind vectors crossing the outer boundary of the CG-MATHEW grid are clipped, vectors are plotted only at internal grid points ( $1 < I < \text{MAT}$  and  $1 < J < \text{JMAXMAT}$ ). This offers some protection against clipping. The density of the vectors can be controlled by giving a value to VECTOR\_PLOT\_FREQ in MEDIC. The default is two (2), which causes vectors to be plotted at every other grid point (see MEDIC description for details). The size of the vectors can be scaled by giving values to the parameter HORZ\_VECTOR\_SCALE in MEDIC (see MEDIC description for details). The levels plotted by default are the reference level,  $1 * \text{DELZ}$ ,  $6 * \text{DELZ}$ , and the upper level winds. All these levels are at constant heights above terrain. The upper level winds are the winds at  $14 * \text{DELZ}$  above terrain, unless the parameterized extrapolation is being used and the top of the boundary layer (BL\_HGT) is greater than  $14 * \text{DELZ}$ . In this case the upper level winds are the winds at the top of the boundary layer. Other model level wind vectors can be plotted by assigning values to PLOT\_LEVELS in MEDIC (see MEDIC description for details).

#### Sample Profile Locations

The positions of the sample profiles are indicated by asterisks (\*) within the CG-MATHEW grid outline. The asterisks are followed by numbers used to identify the profiles.

#### Sample Profiles

The extrapolated winds along a column of CG-MATHEW grid points are displayed as a function of height. Three different profiles are plotted with the vertical extent of each representing the vertical dimension of the grid, ( $14 * \text{DELZ}$ ), above the local terrain. The first display is a profile of speed as a function of height and the second is a profile of direction as a function of height; both profiles are plotted as solid lines. The last display is a vertical line with wind barbs attached. The origins of the wind barbs are placed at various model levels and indicate the extrapolated winds at that level. (See the discussion on “surface data” for a description of wind barbs.) Every other  $k$ -level is shown, as well as the wind at the reference height (REF\_HGT), the top of the surface layer (SL\_HGT), and the top of the boundary layer (BL\_HGT). The speeds and directions at the model grid levels are printed below the displays. If the profile extrapolation procedure is being used, the speed and direction profiles synthesized from the upper air data are shown on the corresponding profile displays as dotted lines. These profiles were used to generate the extrapolated winds along this column of grid points and are helpful in understanding the effects of some of the extrapolation parameters. (See the MEDIC model description for details of the extrapolation process.) If default values are used, plots will be generated for nine columns of grid points spaced evenly over the CG-MATHEW grid area. The user may select different columns by assigning values to IPROFILE and JPROFILE or XPROFILE and YPROFILE in MEDIC (see MEDIC description for details).



## Measured Profiles

Speed and direction profiles and wind barbs are plotted for upper air station observations below the top of the model grid. Asterisks on the profile plots and barbs are shown only at levels for which there is data. If all of the speed or direction data for a station is missing, then the corresponding profile does not appear.

## Input Data

All the input data used in TOPOG and MEDIC is listed where appropriate from the files:

1. TOPOG.NML
2. MEDIC.NML
3. OBSERV.MET
3. STNLOC.MET

### **III.C.8. MATVEL**

MATVEL files are generated by CG-MATHEW and contain the advection (mass-balanced) winds used by ADPIC. Each MATVEL file contains a single data set (i.e., adjusted wind fields created from a single MEDVEL data set). The MATVEL files are binary files used to pass data between codes and is not normally examined by a user. MATVEL file data sets are composed of three data sections:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
UPIC,VPIC,WPIC	IMAXPIC,JMAXPIC, KMAXPIC	REAL*4	ADPIC advection winds

---

These sections are described as follows:

#### Traceback Block

See Appendix C for a description of the traceback block.

#### UPIC, VPIC, WPIC

The advection winds are a subset of the CG-MATHEW grid point adjusted winds chosen to cover the ADPIC grid. These winds are stored in three arrays, one for the  $u$ -components, one for the  $v$ -components, and one for the  $w$ -components of the wind vectors which are expressed as real floating point numbers in m/s. These wind vectors exist at each ADPIC grid point, so each array is dimensioned IMAXPIC by JMAXPIC by KMAXPIC which, for a typical ADPIC grid, is 41 by 41 by 15. These component arrays are stored one after the other ( $u$ ,  $v$ ,  $w$ ) in the MATVEL file data set.

### III.B.9. CG-MATHEW.VLST

CG-MATHEW writes a list of fully qualified, and therefore unique, MATVEL\*.BIF filenames to CG-MATHEW.VLST. This file contains the list of MATVEL\*\*\*.BIF files created by a given CG-MATHEW run. It is read by ADPIC as an optional way of controlling which MATVEL\*\*\*.BIF files to process in a given run. This is an ASCII file that can be edited by a user and provides a mechanism to easily handle multiple MATVEL groups in a single directory by allowing the user to maintain a series of CG-MATHEW.VLST files (under different names with the desired file being copied to CG-MATHEW.VLST as needed).

### III.C.10. CG-MATHEW.LOG

MAT is the ASCII output file for CG-MATHEW. One MAT file is produced for each dataset number, i.e., for each new value of NRUN. This file is used to examine and verify the results of, or to debug, a CG-MATHEWrun. It is not used by other computer codes. During normal execution of the ARAC codestream, this file would rarely be examined, but if problems arise, the information is available as an aid in locating the difficulty. There are as many as nine sections in an MAT file:

1. CG-MATHEW echo
2. Problem data description
3. Grid point extrapolated winds
4. Face-centered extrapolated winds
5. Extrapolated wind divergence
6. Lagrangian multipliers
7. Face-centered adjusted winds
8. Grid point adjusted winds
9. ADPIC adjusted winds

Each of the sections is discussed below:

#### ADPIC Adjusted Winds

The ADPIC wind vectors are printed in the same format as the grid point extrapolated winds, except that an additional array must be written for the  $w$ -components. Also each array is dimensioned IMAXPIC by JMAXPIC by KMAXPIC (41 by 41 by 15 for a typical ADPIC grid). This section will appear only if PRINT\_ADJUSTWIND\_ADPIC is set to 'Y' in the CG-MATHEW file.

#### ADPIC Adjusted Wind Divergence

The ADPIC adjusted wind divergences are printed in the same format as the grid point extrapolated winds. This section will appear only if PRINT\_ADJUSTWIND\_DIVERG is set to 'Y' in the CG-MATHEW file.

### Extrapolated Wind Divergence

The extrapolated wind divergences are printed in the same format as the grid point extrapolated winds. This section will appear only if PRINT\_INITIALWIND\_DIVERG is set to 'Y' in the CG-MATHEW file.

### Face-Centered Extrapolated Winds

The face-centered extrapolated wind components are printed in the same format as the grid point extrapolated winds (see 3 above). This section will appear only if PRINT\_INITIALWIND\_FACE is set to 'Y' in the CG-MATHEW file.

### Face-Centered Adjusted Wind

The face-centered adjusted wind components are printed in the same format as the grid point extrapolated winds, except that an additional array must be written for the  $W$ -components after writing the other two arrays. This section will appear only if PRINT\_ADJUSTWIND\_ADJU is set to 'Y' in the CG-MATHEW file.

### Grid Point Extrapolated Winds

The grid point extrapolated winds are printed as  $u$ - and  $v$ -component arrays of real numbers in m/s. These arrays are printed as a series of two-dimensional arrays, one for each vertical model level. Thus for a typical ARAC grid, 15 arrays, each dimensioned 51 by 51, will be written to define one component. The  $u$ -component arrays are written first, followed by the  $v$ -component arrays. This section will appear only if PRINT\_INITIALWIND\_GRID is set to 'Y' in the CG-MATHEW file.

### Grid Point Adjusted Winds

The grid point adjusted wind vectors are printed in the same format as the grid point extrapolated winds, except that an additional array must be written for the  $w$ -components after writing the other two arrays. This section will appear only if PRINT\_ADJUSTWIND\_GRID is set to 'Y' in the CG-MATHEW file.

### Lagrangian Multipliers

The Lagrangian multipliers are printed, after convergence, in the same format as the extrapolated wind divergence. This section will appear only if PRINT\_LAMBDA is set to 'Y' in the CG-MATHEW file.

### CG-MATHEW Echo

The CG-MATHEW data set is written exactly as it appears in the input file if the CG-MATHEW data was read correctly. Otherwise the line containing the error will be the last to appear.

### Problem Data Description

All parameters and data affecting the problem solution to this point are printed under four headings which indicate the source of the parameters. The subsections are:

- 'TOPOG.NML' -which includes data defined by the terrain data as well as by TOPOG.NML
- 'CG-MATHEW' -which lists the CG-MATHEW parameters, including any default values.

### **III.C.11. CG-MATHEW\*.CGMB**

CG-MATHEW\*.CGMB is the graphics output file for CG-MATHEW. As a default, it is composed of four types of plots:

1. Horizontal plane vector plots
2. Vertical profile locations
3. Vertical wind profiles
4. Input data

These plots are described as follows:

#### Horizontal Plane Vector Plots

Horizontal grids of adjusted CG-MATHEW wind vectors are plotted as constant heights above terrain. These vectors are plotted within the CG-MATHEW grid borders. Since wind vectors crossing the outer boundary of the CG-MATHEW grid are clipped, vectors are plotted only at internal grid points ( $1 < I < \text{IMAX}$  and  $1 < J < \text{JMAX}$ ). The density of the vectors can be controlled by setting a value to VECTOR\_PLOT\_FREQ in CG-MATHEW. The default is two (2), which causes vectors to be plotted at every other grid point (see CG-MATHEW description for details). By default only the CG-MATHEW winds 1\*DELZ above terrain are shown. By selecting model levels through PLOT\_LEVELS and CG-MATHEW, the user can view as many or as few model levels as desired.

#### Vertical Profile Locations

The position of the vertical profiles are indicated by asterisks (\*) within the CG-MATHEW grid boundary. The asterisks are followed by numbers used to identify the profiles.

#### Vertical Wind Profiles

The adjusted wind vectors along a column of CG-MATHEW grid points are displayed as a function of height. Three different displays are plotted, each with the vertical extent representing the vertical range of the CG-MATHEW grid so terrain is included in the plots. The first display is a profile of speed as a function of height and the second is a profile of direction as a function of height. The last display is a vertical line representing the vertical extent of the grid with wind barbs attached at every other model  $k$  level. These barbs represent the adjusted wind at the various levels (see Section 1 of GRAFMEDIC for a discussion of wind barbs). The wind speeds and directions at the model  $k$  levels are printed below the displays. If the default value is used, plots are generated for

nine columns of grid points spaced evenly over the CG-MATHEW grid area. The user may select different columns by assigning values to IPR and JPR in CG-MATHEW (see CG-MATHEW description for details).

### Input Data

All the input data used by TOPOG, MEDIC, and CG-MATHEW is listed where appropriate under four headings:

1. TOPOG.NML
2. CG-MATHEW

### **III.C.12. CONC\*.BIF**

CONC files are generated by ADPIC and contain deposition and instantaneous and integrated air concentration information used by PLCNT in plotting the contours. A CONC FILE is written at each output time and contains five two-dimensional arrays of concentration for each level, up to MAXSBINS levels. Each CONC\*.BIF is a binary file which is rarely examined by the user. It consists of the following data:

<b>Variable</b>	<b>Dimension</b>	<b>Type</b>	<b>Description</b>
Traceback Block			Traceback file information
NUM_BINS	1	Integer*4	Number of sample bins
NUM_NESTS	1	Integer*4	Number of nested grids
X_ADPIC_GRID	IMAXPIC	REAL*4	Model x coords of CELL_CONC array
Y_ADPIC_GRID	JMAXPIC	REAL*4	Model y coords of CELL_CONC array
X_NESTED_GRID	I_MAX_NEST MAX_NEST_DEPTH	REAL*4	Model x coords of CELL_CONC_NESTED arrays
Y_NESTED_GRID	J_MAX_NEST MAX_NEST_DEPTH	REAL*4	Model y coords of CELL_CONC_NESTED arrays
CELL_CONC	IMAXPIC JMAXPIC NUMBERBIN	REAL*4	Concentration arrays
CELL_CONC_NESTED	IMAXPIC JMAXPIC NUMBERBIN MAX_NEST_DEPTH	REAL*4	Nested grid concentration arrays

This variable information is described as follows:

The Traceback Block (TBB) information includes first and last vel file names and portions of the vel file TBB from MEDIC and CG-MATHEW.

### **CELL\_CONC**

These are a series of two-dimensional concentration arrays, where the type of concentration, species sampled, and sampling height have been specified by the user in ADPIC. Up to MAXSBINS) combinations of SAMPLING\_TYPE, SOURCES\_TO\_SAMPLING\_BIN, and SAMPLING\_HGT can be specified in any given run.

### **CELL\_CONC\_NESTED**

These are a series of four two-dimensional nested grid arrays of concentration that correspond to the CELL\_CONC arrays (see Section II.B.3.e. for a detailed explanation).

### **NUM\_BINS**

This variable specifies the number of sample bins

### **NUM\_NESTS**

This is the number of nested grids used by ADPIC.

### **X\_ADPIC\_GRID**

X\_ADPIC\_GRID values are the Model x coordinates of the sampling grid cell centers whose concentrations are given in the CELL\_CONC array.

### **Y\_ADPIC\_GRID**

Y\_ADPIC\_GRID values are the Model y coordinates of the sampling grid cell centers whose concentrations are given in the CELL\_CONC array. When the elements in Y\_ADPIC\_GRID are matched with those in X\_ADPIC\_GRID, the spatial coordinates of the cell centers are completely specified.

### **X\_NESTED\_GRID**

X\_NESTED\_GRID values are the Model x coordinates of the sampling grid cell centers whose concentrations are given in the CELL\_CONC\_NESTED array.

### **Y\_NESTED\_GRID**

Y\_NESTED\_GRID values are the Model y coordinates of the sampling grid cell centers whose concentrations are given in the CELL\_CONC\_NESTED array. When the elements in Y\_NESTED\_GRID are matched with those in X\_NESTED\_GRID, the spatial coordinates of the cell centers are completely specified.

## **III.C..13. ADPIC.LOG**

ADPIC.LOG is a ASCII output file from ADPIC. This file is not used by other codes and it is rarely examined by the user. However, it can serve as a diagnostic aid by helping the user locate the source of a fatal error, when such an error occurs during an ADPIC run. There can be up to six sections in an ADPIC.LOG file:

1. ADPIC.NML Echo
2. New Source Rates
3. Traceback Block
4. Particle Budget

- 5 Particle Summary
6. Optional Output

Each section is discussed below.

#### ADPIC.NML Echo

The ADPIC.NML namelist data is written exactly as it appears in the input file, with default values being applied to all values NOT input in the ADPIC.NML namelist file. This information is also contained in the file ADPIC\*.CNML, part of the traceback system.

#### New Source Rates

Time-varying source rates are incorporated into an ADPIC calculation by using the input array SOURCE\_RATE in ADPIC. A line will be added to the ADPIC.LOG file whenever the source rate is changed; this gives the value of the new rate and the time it is valid.

#### Traceback Block

The Traceback Block contains all the data from the TOPOG.NML, OBSERV.MET, STNLOC.MET, CG-MATHEW, and ADPIC.NML input files. Additionally, a section with the heading MAX VELOCITIES WITH LOCATION INDICES is provided. This section gives the maximum interpolated  $u$ - and  $v$ -component winds with their  $i, j$  and  $k$  indices and the maximum adjusted  $u$ -,  $v$ - and  $w$ -components with their indices. If the particles appear to be moving too fast, a check in this section should indicate whether the problem is caused by advection or diffusion velocities. The next entries are entitled ITER LIMIT and ITER ACTUAL; they define the maximum number of iterations that are allowed for a CG-MATHEW convergence and the actual number of iterations required to adjust the winds for mass consistency.

#### Particle Budget

At the end of each cycle (  $\Delta t$  ), ADPIC write the following information to ADPIC:

- NCYC - Number of the current cycle
- DELT - Time step for this cycle
- TIME - Total time since RUN\_START\_DATE,RUN\_START\_TIME
- NTOT - Total number of active particles
- MGEN - Number of particles generated this cycle
- MTOT - Number of particles generated for entire problem
- LCYC - Number of particles lost this cycle
- LTOT - Total number of particles lost for entire problem
- NOCL - Number of times particles moved more than one cell in any direction
- AVOCL - Average number of cells that NOCL particles moved

If NOCL becomes significantly larger than one, the diffusion velocity algorithm leads to instabilities. The time step can become erratic under these conditions and the particle plots may show a very uneven distribution, especially in the vertical. If this happens, FRACT should be reduced, resulting in a smaller time step, and the problem rerun. FRACT is input through ADPIC.NML; it has a default value of one (1.).

### **Particle Summary**

At the end of each edit (STATUS\_INTERVAL) interval, a partial summary of the particle budget for each source is written to ADPIC. The elements in this list are NTOT, MGEN, and LTOT, and either the particle mass (QSNC) or activity (TACT).

### **Optional Output**

If optional parameters have been selected in ADPIC, they are written to ADPIC.LOG at the end of each edit interval.

#### **III.C.14. ADPIC\*.RST**

ADPIC\*.RST is a binary file which contains all of the information necessary to restart the ADPIC code. The output times (and the permissible restart times) are controlled by the parameter RESTART\_WRITE\_INTERVAL for uniformly spaced ADPIC\*.RST files and with RESTART\_WRITE\_DATES/RESTART\_WRITE\_TIMES arrays for arbitrarily spaced ADPIC\*.RST FILES. These three parameters/arrays can be set in the Adpic.nml file. More specifically, RESTART\_WRITE\_INTERVAL indicates the output interval in hours and minutes ('hhmmss') relative to RUN\_START\_DATE and RUN\_START\_TIME on the first restart, and RESTART\_DATE RESTART\_TIME on subsequent restart. If any values are entered in the RESTART\_WRITE\_DATES/RESTART\_WRITE\_TIMES arrays, those values override the RESTART\_WRITE\_INTERVAL. There is currently a limit of ten restart files per ADPIC run.

#### **III.C.15. CONC\*.RST**

ADPIC accumulates air concentration from the beginning of a run for bins whose sample types are "INTAIR" and dumps this information to CONC\*.RST files each time a restart file (ADPIC\*.RST) is created. The output format of a CONC\*.RST file is identical to the CONC\*.BIF file format, except it contains CELL\_CONC\_CUM and CELL\_CONC\_CUMC\_NESTED data. CONC\*.RST files are read by ADPIC on restart runs and by PLCNT to generate contour plots.



### III.C.16. PARTPOS\*.BIF

Particle position files are generated by ADPIC and can be used directly by graphics post-processing codes to display particle positions or "dots" over terrain images. A PARTPOS file is written at each PLOT\_INTERVAL time interval and contains the following data:

Variable	Dimension	Type	Description
Traceback Block			Traceback file information
format identifier integer	1	Integer*4	1 for particle format, 2 for video
TOTLP_ACTV	1	Integer*4	current total particle count
X_ORIGIN_PIC	1	Real*4	x coord of lower left grid corner
Y_ORIGIN_PIC	1	Real*4	y coord of lower left grid corner
IMAXPIC	1	Integer*4	x grid size by coordinate
JMAXPIC	1	Integer*4	y grid size by coordinate
KMAXPIC	1	Integer*4	z grid size by coordinate
DELX * 1000.	1	Real*4	x grid spacing in meters
DELY * 1000.	1	Real*4	y grid spacing in meters
DELZ	1	Real*4	z grid spacing in meters
HORIGIN	1	REAL*4	grid base elevation in meters
P_X	TOTLP_ACTV	Real*4	x coordinates of particles in meters
P_Y	TOTLP_ACTV	Real*4	y coordinates of particles in meters
P_Z	TOTLP_ACTV	Real*4	z coordinates of particles in meters
VEL_FILE_AGE	1	Real*4	age (hrs) of last velocity data file used
P_SOURCE_NO	TOTLP_ACTV	Integer*4	originating source # for each particle
P_IDENT	TOTLP_ACTV	Integer*4	unique particle identifier number

### III.C.17. ADPIC\*.CGMB

ADPIC\*.CGMB provides three different views of the Lagrangian particle positions as computed by ADPIC at each PLOT\_INTERVAL time. The first plot represents a full projection of the particles onto an  $x, z$  plane, with the plane situated over the Model  $y$  coordinate of the source point (the observer looks from south to north). The second plot represents a projection of the particles onto a  $y, z$  plane positioned over the Model  $x$  coordinate of the source (the observer looks from east to west). Topography is transparent in these displays, but individual boundary cells are frequently outlined, in part, when there are numerous particles in the grid volume. The two plots appear on the same page. Since the aspect ratio of the horizontal and vertical grid dimensions is quite large, the two scales in the vertical have been exaggerated to better view the particle distributions.

### **III.C.18. PLCNT.LOG**

PLCNT.LOG is a ASCII output file from PLCNT. This file is not used by other codes, and it is rarely examined by the user. However, if PLCNT fails to run because of an improper input, then PLCNT.LOG can be helpful in locating the source of the fatal error. There can be up to two sections in the PLCNT.LOG file:

1. PLCNT Echo
2. Concentrations

Each section is described below:

#### Concentrations

The two-dimensional arrays of grid point concentrations which are used by PLCNT in providing the contour plots are written to PLOT. Each array is preceded by all of the descriptor information that appears on the corresponding plot. (See Section III.C.14. for further details.)

#### PLCNT Echo

These data are written exactly as they appear in the PLCNT file, provided PLCNT has been read correctly. If not, the last line will contain the error.

### **III.C.19. CENLINE\*.DAT**

CENLINE\*.DAT is a ASCII output file generated from a centerline run of PLCNT. The primary contents of this file are tables of maximum concentrations and their locations (distance in km and bearing from source location, as well as the x and y coordinates)

### **III.C.20. TIMEHIS.LOG**

TIMEHIS.LOG is a ASCII output file from TIMEHIS. This file is not used by other codes, and it is rarely examined by the user. However, if TIMEHIS fails to run because of an improper input, then TIMEHIS.LOG can be helpful in locating the source of the fatal error. There can be up to four sections in the TIMEHIS.LOG file:

1. TIMEHIS Echo
2. Concentrations
3. Samplers

Each section is described below:

#### Concentrations

The two-dimensional arrays of grid point concentrations which are used by TIMEHIS in providing the contour plots are written to TIMEHIS. Each array is preceded by all of the descriptor information that appears on the corresponding plot. (See Section III.C.14. for further details.)

#### TIMEHIS Echo

These data are written exactly as they appear in the TIMEHIS file, provided TIMEHIS has been read correctly. If not, the last line will contain the error.

#### Samplers

If the user decides to place samplers within the ADPIC grid, then the sampler names and their values are printed after the corresponding concentration array. After writing the final set of concentrations to TIMEHIS, TIMEHIS will then output a list of the sampler values.



## APPENDIX A. BIBLIOGRAPHY

- Amsden, A.A., 1966: The particle-in-cell method for the calculation of the dynamics of compressible fluids. Los Alamos Scientific Laboratory Report LA-3466.
- Batchelor, G.K., 1947: Kolmogoroff's theory of locally isotropic turbulence. *Proc. of Cambridge Phil. Soc.*, **43**, 533.
- Batchelor, G.K., 1950: The application of the similarity theory of turbulence to atmospheric diffusion. *Quart. J. Roy. Meteor. Soc.*, **76**, 133.
- Boughton, B.A., and J.M. Delaurentis, 1987: An integral model of plume rise from high explosive detonations. 1987 ASME/AIChE National Heat Transfer Conference, August 9–12, Pittsburgh, PA.
- Briggs, G.A., 1972: Plume rise predictions. Lectures on Air Pollution and Environmental Impact Analyses (Ed. Haugen), *American Meteorological Society*, pp. 53–111, Boston MA.
- Businger, J.A., J.C. Wingard, Y. Izumi and E.F. Bradley, 1971: Flux profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, **28**, 181.
- [Courant, R. and D. Hilbert, 1962: \*Methods of Mathematical Physics\*, Interscience Publishers, John Wiley and Sons, New York.](#)
- Crandall, W.K., C.R. Molenkamp, A.L. Williams, M.M. Fulk, R. Lange, and J.B. Knox, 1973: An investigation of scavenging of radioactivity from nuclear debris clouds: Research in Progress. Lawrence Livermore National Laboratory Report UCRL-51328.
- Crawford, T.V., 1966: A computer program for calculating the atmospheric dispersion of large clouds. Lawrence Livermore National Laboratory Report UCRL-50179.
- Crawford, T.V., 1974: Progress Report Dose-to-Man Program FY-1973. E.I. duPont de Nemours and Co., Savannah River Laboratory, Aiken, SC, NTIS DP-1341, 53–61.
- Davenport, A.G., 1965: The relationship of wind structure to wind loading. National Physical Laboratory Symposium No. 16, *Wind Effects on Buildings and Structures*, pp. 54–102, Her Majesty's Stationery Office, London.
- DeMarco, T., 1979: *Structured analysis and system specification*. Yourdon, Inc., New York, NY.
- Desiato, Franco and Rolf Lange, 1986: Introduction of plume rise capability into the transport and diffusion codes of the ARIES/ARAC emergency response systems. Lawrence Livermore National Laboratory Report UCID-20652.
- Dickerson, M.H. and P.H. Gudiksen, 1980: ASCOT FY-1980 Progress Report. Lawrence Livermore National Laboratory Report UCID-18878, ASCOT 80-10.
- Draxler, R.R., 1976: Determination of atmospheric diffusion parameters. *Atmos. Environ.*, **10**, 99–105.
- Forsythe, G.E., and W.R. Wasow, 1960: *Finite-difference methods for partial differential equations*. John Wiley Sons, Inc., New York, NY.
- Environmental Protection Agency (EPA), 1981, "Preface" in *Summary Report of the Regional Workshops on Air Quality Modeling*, EPA, Washington, D.C.
- Foster, K.T. and M.H. Dickerson, 1990: An updated summary of MATHEW/ADPIC Model Evaluation Studies. Lawrence Livermore National Laboratory Report UCRL-JC-104134.
- Foster, K.T., R. Freis, J.S. Nasstrom, 1990: Incorporation of an explosive cloud rise code into ARAC's ADPIC transport and diffusion model. Lawrence Livermore National Laboratory Report UCRL-ID-103443.
- Golder, D., 1972: Relationships among stability parameters in the surface layer. *Boundary Layer Meteorology*, **3**, pp.47–58.
- [Gresho, P., 1975: \*UASG\*, 75–22.](#)
- Holzworth, G.C., 1964: Estimates of mean maximum mixing depth in the contiguous United State. *Mon. Wea. Rev.*, **92**, (5), 235.
- Lange, R., 1973: A three-dimensional computer code for the study of pollutant dispersal and deposition under complex conditions. Lawrence Livermore National Laboratory Report UCRL-51462.

- Lange, R., 1978: ADPIC—A Three-Dimensional Particle-in-Cell Model for the Dispersal of Atmospheric Pollutants and Its Comparison to Regional Tracer Studies. *J. Appl. Meteor.*, **17**, 320–329.
- Lange, R., 1989: Transferability of a three-dimensional air quality model between two different sites in complex terrain. *Journal of Applied Meteorology*, **Vol 28**, 7, pp. 665–679. Also Lawrence Livermore National Laboratory Report UCRL-96220.
- McDonald, J.E., 1960: An aid to computation of terminal fall velocities of spheres. *J. Meteor.*, August 1960, p. 463.
- McMahon, T.A. and P.J. Denison, 1979P: Empirical atmospheric deposition parameters—A survey. *Atmospheric Environment*, **13**, pp. 571–585.
- Monin, A.S. and A.M. Obukhov, 1954: Basic laws of turbulent mixing in the ground layer of the atmosphere. *Akad. Nauk, USSR Geofiz. Ins. Tr.*, 151, 163.
- Page-Jones, M., 1980, *The Practical Guide to Structured Systems Design*, Yourdon Press, New York, NY.
- Pasquill, F., 1971: Atmospheric dispersion of pollution. *Quart. J. Roy. Meteor. Soc.*, **97**, 369.
- Press, W., S. Teukolsky, W. Vetterling, and B. Flannery, 1993: *Numerical Recipes*, 2nd. ed., Cambridge University Press, Cambridge, U.K.
- Sasaki, Y., 1958: An Objective Analysis Based on the Variational Method. *J. Meteor. Soc. Jap.*, **36**, 77.
- Sasaki, Y., 1970a: Some basic formalisms in numerical variational analysis. *Mon. Wea. Rev.*, **98**, 875.
- Sasaki, Y., 1970b: Numerical variational analysis formulated under the constraints determined by longwave equations and low-pass filter. *Mon. Wea. Rev.*, **98**, 884.
- Sherman, C.A., 1978a: MATHEW: A mass-consistent wind field model. Ph.D. Thesis, University of California, Davis/Livermore. Lawrence Livermore National Laboratory Report UCRL-52479.
- Sherman, C.A., 1978b: A mass-consistent model for wind fields over complex terrain. *J. Appl. Meteor.*, **17**, 312–319.
- Sklarew, R.C., A.J. Fabrik and J.E. Prager, 1971: A particle-in-cell method for numerical solution of the atmospheric diffusion equation, and application to air pollution problems. Systems, Science and Software, La Jolla, CA, Report 3SR-844.
- Slade, D.H., 1968: Meteorology and atomic energy. U.S. Atomic Energy Commission, Technical Information Division, Oak Ridge, TN.
- Start, G.E., C.R. Dickson and L.L. Wendell, 1975: Diffusion in a canyon with rough mountainous terrain. *J. App. Meteor.*, **14**, 333–346.
- Sugiyama, Gayle, Robert E. Lee, and Hoyt Walker, 1994: Conjugate Gradient MATHEW. Lawrence Livermore National Laboratory, UCRL-ID-118629.
- Varga, R., 1962: *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, New Jersey.
- Walton, J.J., 1973: Scale dependent diffusion. *J. Appl. Meteor.*, **12**, 547–549.
- Weil, J.C. and D.P. Hoult, 1973: A correlation of ground level concentrations of sulfur dioxide downwind of the keystone stacks. *Atmospheric Environment*, **7**, pp. 707–721.
- Welch, J.E., F.H. Harlow, J.P. Shannon and B.J. Daly, 1965: The MAC Method. Los Alamos Scientific Laboratory Report LA-3425.
- Yourdon, E. and L.L. Constantine, 1978: *Structured Design*, Yourdon Press, New York, NY.

## **APPENDIX B. TOPOG INTERACTION**

The interactive capabilities in TOPOG provide a means to customize the model topography to the needs of a problem when the default elevation data averaging in TOPOG produces less than ideal results. These interactive capabilities are a combination of graphical presentations of the elevation data (color coded cells, contours, and cell values for a subset of the grid) and SmartStar menus/forms that permit the user to select the desired options. There are three menus in TOPOG: the MAIN MENU, the UNITS MENU, and the EDIT MENU. The main menu is the top level in the hierarchy with the other two menus accessed from the MAIN MENU. The UNITS MENU is selected via the 'Change modes' option on the MAIN MENU while the EDIT MENU is also selected via the 'Edit subgrid' option with the additional step of selecting a region on the display for more detailed examination with a graphics input device, typically a mouse. These menus are shown below.

### **TOPOG MAIN MENU**

1. Edit subgrid
2. Fill holes
3. Change modes
4. Plot contours
5. Plot cell levels

### **TOPOG UNITS MENU**

1. Units to DELZ
2. Units to METERS
3. Units to FEET
4. Set contour interval
5. Set DELZ
6. Reset grid base

### **TOPOG EDIT MENU**

1. Edit single cell
2. Edit subgrid row
3. Edit subgrid column
4. Edit CG-MATHEW row
5. Edit CG-MATHEW column
6. Undo last edit

The follow is a discussion of the use of TOPOG interactive capabilities taken almost entirely from a series of documents written by John Pace. As these provide an excellent and complete description of the thoughtful use of these tools, his work is included here adapted slightly for the CG-MATHEW/ADPIC User's Guide. First, each option in the TOPOG menus is discussed. After this a good example of the use of these features is provided.

When you first run TOPOG, you get a cell-level map of the entire CG-MATHEW grid area on the display screen, and the MAIN MENU on the control screen.

**The MAIN MENU contains the following options:**

1. Edit subgrid. Selecting the "Edit subgrid" option from the TOPOG MAIN MENU means you want to edit a 10X10 subgrid of the full CG-MATHEW grid. So, you have to tell the system where the subgrid is that you want to edit. On the full display map, you left-click where you want the center of your subgrid to be, and then (after the display map is redrawn, for reasons related to the GKS graphics standard) you left-click again to verify you really wanted the subgrid where you said you did. If you change your mind, you can select a different subgrid center by center-clicking. (If you put the center of the subgrid close to the edge of the CG-MATHEW grid, the full size subgrid is built, but TOPOG invents subgrid cells for the area not covered by the CG-MATHEW grid, with fictitious values in the subgrid cells that aren't really there.) The system draws your 10X10 subgrid. Then, when you switch to the control screen and hit return, you get the EDIT MENU, which is described below.

2. Fill holes. This is the fastest and easiest way to finish TOPOG and go on to the rest of the problem. You must be in DELZ mode to use this option (see the Change modes option).

3. Change modes. Enter the UNITS MENU.

4. Plot contours and use this display of the elevation data as the default. You can control the contouring interval (CONTOUR\_INTERVAL) and/or the model's vertical grid spacing (DELZ) in the TOPOG.NML file, or interactively by using the Change modes option. This option lets you see a contoured display of the terrain field of the entire CG-MATHEW grid. This function is controlled by the CONTOUR\_INTERVAL and DELZ parameters, which can be set in the TOPOG.NML file or modified as discussed above, under the "Change modes" option. The contouring graphics package calculates how many contours it needs to use, with the interval set by CONTOUR\_INTERVAL, to build the entire depth of the model grid (usually, 14 times DELZ). The package uses only 7 colors, so if it needs more than 7 contour intervals it assigns several consecutive contour intervals to the same color. Thus it's possible to see several green contours, for example, surrounding a yellow contour. Once you've created a contour map, it's quick and easy to redraw the map using new contour control parameters (CONTOUR\_INTERVAL and DELZ). From the MAIN MENU, select "Change modes" to get to the UNITS MENU. Then you can change "Set contour interval" and/or "Set DELZ". Hit GOLD/ZERO and you'll get a new contour map. It's a rather unusual graphics effect as the old map melts away--give it a try.



5. Plot cell levels and use this display of the elevation data as the default. This will return the default display map to what you started with when you first ran TOPOG.

**The EDIT MENU contains the following options:**

1. Edit single cell. This option lets you change the elevation value of just one subgrid cell. (This was the main subject of yesterday's note.) If you're in the DELZ mode, you assign a new elevation in DELZ units. If you're in the METERS or FEET modes, you assign a new actual elevation. When you select this option, you must tell TOPOG which cell in the subgrid is to be edited. So, left-click on the elevation value in the cell to be changed. The subgrid is redrawn with the elevation of that cell in yellow. Left-click again, somewhere on the subgrid, to verify your selection. Click on the control screen. Enter a new cell elevation value, in the units set by the mode you're in, and hit return. The elevation value of the cell is changed and is now displayed in white. Continue editing this subgrid by hitting keys 1-5, or hit GOLD/ZERO to get back to the MAIN MENU with the change retained, or hit 6 to cancel the edit and get back to the EDIT MENU.

2. Edit subgrid row. This option lets you edit all 10 cells in a subgrid row. You must tell TOPOG which row to edit. Left-click on an elevation value in any cell in the row you want to change. The subgrid is redrawn, with the values in all the cells in that row changed to yellow. Left-click again to verify your selection. Switch to the control screen. Enter the new values, in the units you're working in (DELZ units, meters, or feet), in the 10 available spaces. These new values go into the subgrid cells in your row, from left to right, when you hit GOLD/ZERO. You'll see the new values in white. Continue editing this particular subgrid, or hit GOLD/ZERO again to get back to the MAIN MENU and retain the change, or hit 6 to cancel the edit and get back to the EDIT MENU.

3. Edit subgrid column. Pretty much the same as "Edit subgrid row", except of course you're selecting a particular column of values to edit. When you enter new values on the control screen, keep in mind they'll be inserted from bottom to top of the subgrid.

4. Edit CG-MATHEW row. This option lets you enter values for all the cells in a row of the CG-MATHEW grid (usually 50 of them). You can see only 10 of the grid cells in your subgrid, so you shouldn't try to use this option for the purpose of eliminating holes. However, if you need to enter values for a grid, when you don't have terrain data to work with but must instead get elevations from a topographical map, this is the way to go.

5. Edit CG-MATHEW column. Same as Edit CG-MATHEW row for a column.

6. Undo last edit. As mentioned above, this option erases the effect of the last edit you made, and returns you to the EDIT MENU.

**The UNITS MENU contains the following options:**

1. Units to DELZ. When you set up the TOPOG.NML file, you either define DELZ explicitly as the vertical grid spacing, or you define the complete vertical depth of the grid (Z RANGE) and let TOPOG calculate DELZ for you by dividing Z RANGE by the number of vertical grid spaces (usually 14). Once DELZ is known, TOPOG sets up categories of DELZ units. For example, if DELZ=50 meters, the units of DELZ correspond to various height categories as follows:

DELZ =	1	2	3	4	etc.
ELEV Range (m) =	0-24	25-74	75-124	125-174	

A useful thing to remember is that if you've changed the mode to something other than DELZ, you can't select "Fill holes" from the MAIN MENU until you've changed the mode back to "Units to DELZ".

One problem with the DELZ units system is that it doesn't reveal much about the detailed vertical structure. As I discussed yesterday, you can see a lot more by switching the mode to METERS (or FEET). TOPOG defines one-cell holes as cells where the DELZ value is less than the DELZ value in the four surrounding cells. So, using the DELZ mode can help you see where holes may still exist, even after you've done some editing. Also, since TOPOG defines its output terrain in DELZ units, you can use the DELZ mode to see what TOPOG considers to be its output. After you've selected "Units to DELZ", hit GOLD/ZERO to return to the MAIN MENU.

2. Units to METERS. If you select this mode (and hit GOLD/ZERO to return to the MAIN MENU), you'll be able to see the actual elevation in meters that TOPOG has assigned to each cell when you edit a subgrid. And, when you enter new subgrid cell elevation values, they'll be in meters, not in DELZ units. It's not difficult to set up a table relating DELZ and the actual elevation in meters, as I did above, so it's easy to be sure you're changing heights enough to result in a new DELZ

value, and thereby eliminate holes. Keep in mind that TOPOG will convert all heights into DELZ units before creating its output files, so you can't create more resolution than DELZ provides. But the advantage of the "Units to METERS" option is that it lets you see what you need to know, so you can fill or expand holes effectively.

3. Units to FEET. Compared to "Units to METERS", this mode provides a little more detailed information about the actual elevation in each subgrid cell. However, it's harder to use, because it's harder to relate DELZ units (which are based on meters) to feet. So, it's less clear whether or not you've changed a cell's height enough to get it into the next DELZ category. Its main function is to facilitate in the input of elevation data directly from a contour map where the contour elevations are in feet.

4. Set contour interval. This value has no effect on the actual execution of the model. Its effect is only on the optional display you can get by selecting "Plot contours" from the MAIN MENU (see below). This option effectively allows you to override the choice of CONTOUR\_INTERVAL you made in TOPOG.NML. Or, if you didn't define CONTOUR\_INTERVAL in TOPOG.NML, TOPOG will use the DELZ value as the contour interval unless you use the "Set contour interval" option. Regardless of whether you're in DELZ, METERS, or FEET mode, when you set the contour interval you're entering a value in meters!

5. Set DELZ. In contrast to the "Set contour interval" option, this option has a strong effect on the actual execution of TOPOG. It lets you change the actual vertical structure of the model grid! If you change DELZ, TOPOG will recalculate all the results that depend on vertical structure, including which cells are now holes. As with "Set contour interval", the value you enter for DELZ is always in meters, regardless of what mode you're working in.

6. Reset grid base. Substantial editing of the cell heights can cause the base of the model topography to be offset. The grid base is adjusted during the averaging process so the bottom of the grid volume is at the zero cell height index (corresponding to an index of one in MEDIC and CG-MATHEW). If editing causes the lowest level of the model topography to be raised above zero, then the volume of the grid above the terrain is reduced unnecessarily. The 'reset grid base' option permits the grid base to be shifted so the lowest index is again zero. An important example of the use of this option occurs when a complete elevation grid is input interactively based on values read from a contour map. Typically, TOPOG is initiated with FLAT\_TOPO set to 'Y' setting the initialize cell heights to zero. The grid base is also set to zero. After the cell heights have been entered, the grid base remains at zero, while the cell heights may all extend beyond the grid top (e.g., consider entering elevations from the Himalayas with a 100 m DELZ producing a minimum cell height of,

perhaps, 40 or 50). In such a case, 'reset grid base' must be performed to produce a grid that includes some atmosphere.

### **Interactive example:**

As mentioned in Section II.C.1, single cell holes in the model topography must be avoided. These can be removed automatically, either by running TOPOG with INTERACTIVE set to 'N' or by selecting option 2 in the MAIN MENU. A second way to eliminate holes in the initial TOPOG terrain field, that in some cases may be better than the default hole filling, is to edit the cell heights interactively. This approach lets you eliminate holes while changing the terrain field as little as possible, and/or retaining as much as possible the significant features of the terrain. The drawback is that it takes longer.

The terrain files have 1-meter-resolution elevation data, and TOPOG's job is to convert these values into representative heights, in multiples of DELZ, for each CG-MATHEW/ADPIC grid cell. For example, if you've set DELZ=50 m, the units of DELZ correspond to heights in various categories, as shown:

DELZ =	1	2	3	4	etc.
ELEV Range (m) =	0 - 24	25 - 74	75 - 124	125 - 174	
Avg Elev (m) =	12	50	100	150	

So, if the elevation in a grid cell is between (and including) 25 and 74 meters, TOPOG will assign it a DELZ value of 2.

A one-cell hole has a DELZ value (corresponding to these ranges of actual elevations) at least one DELZ unit less than the values in the four rook's case neighboring cells. If you let TOPOG fill the hole automatically, it will increase the hole's DELZ value so it matches the lowest elevation in any of the surrounding cells.

You have the capability to fill holes manually. If you choose to do this, you'll see the system is set up to show each cell's elevation in DELZ units, with one-cell holes identified as red squares. You have two ways to eliminate one-cell holes: by increasing the cell-hole's DELZ value to match or exceed the surrounding values; or by reducing the height of an adjacent grid cell to create a larger valley than you had before. Either approach satisfies ADPIC. But this configuration gives you no way to know which of these options is better (more faithful to the actual terrain) in a given situation. However, TOPOG can provide additional information to assist in resolving the ambiguity. The

assessor can look at the actual elevation in meters in each cell, and make an educated decision as to whether the hole should be filled or the valley should be expanded.

To do this alternate procedure, do the following:

a. Start TOPOG in the ordinary way. You'll see the usual map, with black squares showing the locations of all the holes.

b. On the MAIN MENU, select item 3 (Change modes). On the UNITS MENU, select item 2 (Units to METERS). Then press GOLD ZERO to get back to the MAIN MENU. Now press 1 (Edit subgrid). Move the cursor near the hole you want to eliminate, and left-click. After the screen settles down, left-click again to accept that location (or center-click if you want to pick another location). The screen is redrawn as a 10 X 10 subgrid, with each cell's elevation shown in meters. Now you have the information you need to make the best choice of whether a hole will be filled or expanded.

c. You make the change by altering one or more cell's elevation so it crosses into the next DELZ category (either up or down). To do this, go back to the control screen. The EDIT MENU should be active. Select item 1 (Edit single cell). Left-click your cursor on the cell you want to change. When the screen settles down, left-click again to verify your choice. You'll see the elevation value is now displayed in yellow. Now click on the control screen, and enter the new elevation value for the cell. The new elevation is shown in white on the subgrid cell. Hit GOLD ZERO to exit, retaining the change. If you've eliminated one or more holes, you'll see that reflected in the screen display.

Here's an example. I made a TOPOG run which initially had 11 one-cell holes in the grid. The grid cells in the vicinity of one of the holes had these elevations (in meters):

54	58	32	55	52	33
63	57	18	49	58	28
67	46	25	49	44	32
62	28	43	38	19	42

The one-cell hole was the "18" value on the second line. Notice that there was a "25" value immediately south of the hole, so there actually was an area of lower elevation in these two cells, surrounded by higher terrain. If the "25" was instead a "24", TOPOG wouldn't think this was a one-cell hole at all. (The "19" value on the fourth line was north of another lower value, so it wasn't a one-cell hole.)

The DELZ values corresponding to this initial field were:

2	2	2	2	2	2
2	2	1	2	2	2
2	2	2	2	2	2
2	2	2	2	1	2

If I had let TOPOG fill the holes automatically, the "1" on the second line would have become a "2", and the resulting elevation field wouldn't have reflected the lower terrain in that area. On the other hand, it was easy to change the "25" value to "24" (while in the meters mode), eliminating the one-cell hole while retaining the lower-elevation feature. (Or, I could have changed that cell's "2" to a "1" while I was in the DELZ mode). The point is that if you don't use the meters mode, you won't know which approach will best preserve the terrain's characteristics. And, of course, if you let TOPOG fill holes automatically you can't make these decisions at all. This hands-on procedure takes longer than letting TOPOG do it automatically. So, perhaps you might want to use it only for holes close to the expected path of the plume. Then you can let TOPOG fill the rest of the holes on its own.

## **APPENDIX C. TRACEBACK BLOCK DESCRIPTION**

### Traceback block structure

Historically, the transfer of parameters and the traceback functions was implemented via an ID block, which was intended to be a sequential list of all model parameters and meteorological data used in a model execution and was placed at the beginning of each file used to transfer data between models. Each such file contains space for a complete ID block which with appropriate sections being filled in as the model sequence was executed. As a result, much of the space was not utilized until the completion of ADPIC.

While the ID block functioned for 15+ years, its use implied a number of constraints that limited the development of the models and increased the difficulty of maintenance. Many of these constraints were associated with the "hard-wired" aspects of this mechanism that forced recompilation of the entire model stream to make any change in the ID block structure along with an inability to use any binary files generated with an old ID block. As a result, there was a strong tendency to avoid changes in the modeling system. Such inertia is inappropriate for an active model development environment. Therefore, it was necessary to replace the ID block with a new model environment that supports the necessary parameter passing and traceback while permitting easier changes to the models.

The traceback block (TBB) is composed of series of sections, one for each model plus an additional section for general pointers that define the modeling environment and a one entry section giving the name of the current file. The location of individual entries within a TBB section is specified by parameters defined in an INCLUDE file as follows:

TBB\_SELF.CON

TBB\_GEN.CON

TBB\_TOP.CON

TBB\_MED.CON

TBB\_MAT.CON

TBB\_ADP.CON

TBB\_PLC.CON

TBB\_TIM.CON

New files requiring traceback support are to be added at the end of the appropriate section (the mechanism for defining section lengths is discussed under the next heading in this document). Thus, the order of traceback entries, once defined, must remain unchanged to support the use of older data files with newer models. Adding a new file entails updating the INCLUDE file and recompiling the models that make use of the TBB section. An unused entry will be maintained at the end of each section to support testing of a model with an additional file without recompiling other models that don't read the new file.

The following list is the current structure of the TBB in order. Each entry is a 40 character string. Note that directory names are assigned two entries. Also note that .PRM extensions indicate files in the Model Parameters System:

Self 1    current file name

#### General

- 1    MODEL\$A\_GRID
- 3    MODEL\$A\_WINDS
- 5    MODEL\$A\_DISPERSION
- 7    MAP\$A\_REGIONAL
- 9    MAP\$A\_PROJECTION\_PARAMETERS
- 10   SITE\_ID
- 11   PROBLEM\_ID
- 12   TOPOG\_GRID
- 13   MAP\$F\_SITEMAP
- 14   MAP\$F\_PROJECTION\_PARAMETERS
- 15   Unused Entry
- 16   Unused Entry

#### TOPOG

- 1    TOPOG.NML.CNML
- 2    TOPOG.PRM
- 3    TOPO\$A\_REGIONAL
- 5    TOPO\$F\_REGIONAL
- 6    unused entry

#### MEDIC

- 1    MEDIC.CNML



- 2 STNLOC.CMET
- 3 OBSERV.CMET
- 4 MEDIC.PRM
- 5 REGMET.PRM
- 6 GWC Grid directory
- 8 GWC Grid
- 9 RWM Grid directory
- 11 RWM Grid
- 12 unused entry
- 13 unused entry

#### CG-MATHEW

- 1 CG-MATHEW.CNML
- 2 CG-MATHEW.PRM
- 3 REGMET.PRM
- 4 MEDVEL.BIF
- 5 unused entry

#### ADPIC

- 1 ADPIC.CNML
- 2 ADPIC.RST
- 3 ADPIC.PRM
- 4 REGMET.PRM
- 5 First MEDVEL.BIF/MATVEL.BIF
- 6 Last MEDVEL.BIF/MATVEL.BIF
- 7 ADPIC.VLST
- 8 HYBRID\_DB
- 9 ADPIC.RST
- 10 CONCRST
- 11 RESUS.BIF
- 12 unused entry
- 13 unused entry

#### PLCNT

- 1 PLCNT.CNML
- 2 CONTOUR.PRM

- 3 DOSECON.PRM
- 4 First CONC.BIF
- 5 Last CONC.BIF
- 6 PLCNT.CLST
- 7 unused entry

#### TIMEHIS

- 1 TIMEHIS.CNML
- 2 TIMEHIS.PRM
- 3 DOSECON.PRM
- 3 SAMPLER.CDAT
- 4 First CONC.BIF
- 5 Last CONC.BIF
- 6 TIMEHIS.CLST
- 7 unused entry

Note that the first three entries in the general section are three logical names that control the directories where data files are found. MODEL\$A\_GRID specifies the directory where the TOPOG\*.GRID file is found. MODEL\$A\_WINDS indicates the location of the data used by MEDIC and CG-MATHEW. And MODEL\$A\_DISPERSION shows where ADPIC, PLCNT and TIMEHIS data can be found. This permits reuse of previously created TOPOG\*.GRID and MED/MATVEL files without recopying the files to a new run directory. Also note that this same structure is used for the hemispheric models that are essentially a parallel model stream in terms of input files with the exception of HMEDIC.

#### Traceback block structure definition

It can be assumed with some confidence that the ARAC models will continue to require new data files over time in order to support new data sources and more sophisticated modeling functions. These new files will require traceback support. To facilitate such support and to avoid incompatibilities in the use of old model files, a capability has been developed to allow old model files to describe their traceback structure via metadata at the beginning of the TBB. This involves maintaining a file listing the current number of entries in each TBB section

(TBB\_STRUCTURE.CON). This information is placed at the beginning of each .GRID and .BIF file thus defining the TBB structure that was in use when the file was created. The structure definition includes the number of sections in the TBB, thereby permitting addition of new models and other programs to the traceback system. When an old file is read, the metadata will allow later codes to determine the location of the necessary filenames inspite of the out-of-date TBB structure.

All the models that read the TBB have buffers of TBB entries that are long enough to hold the current TBB plus room for a substantial amount of growth to minimize the need to recompile the codes. As the ARAC models develop, these buffers will gradually be used more completely. If they are ever nearly full, they would have to be expanded; thus, they can never get shorter only longer. Increasing the buffer size may be an inconvenience but it should be simple to perform and will not cause an inability to read and process old data files.

Because the various buffers only grow with time and each .GRID and .BIF file defines its own TBB structure, any files created using this system will be readable by any future version of the CG-MATHEW/ADPIC models for which the basic model and grid structure remains consistent (a terrain-following coordinate CG-MATHEW/ADPIC or a finite-element SABLE/ADPIC would presumably imply an unavoidable discontinuity in the file handling system). Thus, it should be possible to use an old .GRID file with a antiquated TBB and wind grids using a more recent TBB with the latest version of ADPIC without difficulty. The ease of updating the TBB structure suggests that it is not necessary to leave large amounts of empty space within the current TBB to support future model development, although the TBB buffers within the codes should support an appropriate amount of extra space.

A sample TBB header follows:

Self ID  
General  
TOPOG  
MEDIC  
CG-MATHEW  
ADPIC  
PLCNT  
TIMEHIS



## APPENDIX D. ARAC Model System Logical Names

VAX VMS logical names play a central role in the ARAC modeling system and it is important for model users to have some familiarity with these tools (it is intended that most or all of the functionality implemented in VMS with logical names will be mapped to environment variables in Unix versions of the system). Logical names are used to point at all directory paths in file handling operations and to some files. In doing this, flexibility is provided in locating input and output files in a directory structure that would not be possible if the disk/directory paths were set explicitly in the codes. There are several logicals that are used throughout the model system. In addition, TOPOG, MEDIC and PLCNT have a number of additional logicals that define connections to various parts of the system.

### General Logicals

There are four logicals that are central to the execution of the ARAC models. The first three control the directories where input and output from the various codes is accessed. MODEL\$A\_GRID is associated with TOPOG, MODEL\$A\_WINDS is associated with MEDIC and CG-MATHEW, and MODEL\$A\_DISPERSION is associated with ADPIC, PLCNT and TIMEHIS. More specifically, TOPOG reads its control input (TOPOG.NML) and writes the TOPOG\*.GRID file to the MODEL\$A\_GRID directory. All subsequent models also read the TOPOG\*.GRID file from this directory. Because this directory specification is a logical name the various models can read the TOPOG\*.GRID file from another directory produced as part of a different run thereby avoiding the need to copy the TOPOG\*.GRID file from run directory to run directory to execute different model runs using the same grid geometry.

MEDIC and CG-MATHEW read their control input (MEDIC.NML, OBSERV.MET, STNLOC.MET, CG-MATHEW.NML and MEDIC.VLST) and write their output to MODEL\$A\_WINDS. ADPIC, PLCNT and TIMEHIS read their control input (ADPIC.NML, CONC\_ANALYSIS.NML, ADPIC\*.CLST and SAMPLER.DATA) and write their output to MODEL\$A\_DISPERSION. These logicals are set by the restructured model version of SELRUN in the operational system and in the exem environment. *These logicals must be set properly for the system to run correctly.*

The fourth essential logical is MODEL\$F\_GRID that defines the location of the TOPOG\*.GRID file. This logical typically includes MODEL\$A\_GRID: followed by the fully qualified TOPOG\*.GRID filename. MODEL\$F\_GRID is defined by TOPOG when the code is executed. Thus, in a straightforward execution of the models, TOPOG is run, followed immediately by MEDIC with no operation being required to set this logical name. When returning to a directory or when a TOPOG\*.GRID file is to be used from a different run directory, then the SELGRI command is run

that allows the user to select a MODEL\$A\_GRID directory along with the appropriate TOPOG\*.GRID file. *This logical must also be set for the system to run correctly.*

Other logicals in general use are GEOG\$A\_REGIONAL and GEOG\$F\_REGIONAL that point to the geography file that is to be used as a base map for various ARAC model plots. GEOG\$A\_REGIONAL points at the directory that contains a map family while GEOG\$F\_REGIONAL consists of GEOG\$F\_REGIONAL: followed by the specific map file name. These logicals are not essential to the running of the models, if a map file cannot be opened and read successfully, the models just don't plot a basemap. The ARAC system logicals SITE\_ID, PROBLEM\_ID and RUN\_ID are translated and placed in the traceback block to provide a record of the run environment when a file was generated.

### **TOPOG Logicals**

TOPOG relies on two logicals that define the elevation data to be used. These are TOPO\$A\_REGIONAL and TOPO\$F\_REGIONAL with GEOG\$A\_REGIONAL point to the directory that contains the elevation data files and TOPO\$F\_REGIONAL consisting of TOPO\$F\_REGIONAL: followed by the specific elevation file name. These logicals must be set if FLAT\_TOPO is 'Y' in TOPOG\_CONTROL\_NML but are not used if FLAT\_TOPO is 'N'.

### **Map Projection Logicals**

The models are executed on the current map projection as defined by the logical MAP\$F\_PROJECTION\_PARAMETERS. This logical points at an ASCII file that defines a projection. The pathname for the projection file is defined by the logical MAP\$A\_PROJECTION\_PARAMETERS. Translations of both these logicals are entered into the traceback block by TOPOG when a .GRID file is created. Note that the value in MAP\$A\_PROJECTION\_PARAMETERS and MAP\$F\_PROJECTION\_PARAMETERS apply problem-wide, that is, all runs in a problem use the same map projection. Outside of the operational system, the user must ensure that the MAP\$F\_PROJECTION\_PARAMETERS and MAP\$A\_PROJECTION\_PARAMETERS logicals are set to point at the appropriate projection definition file.

### **MEDIC Logicals**

MEDIC utilizes a number of logicals in order access meteorological information from various sources. The problem station library is the automated source of meteorological observing station locations (thus, superceding the STNLOC.MET file) and is accessed using the logical EDITPSL\$F\_PROBLEM\_STATION\_LIBRARY. This information may also reside in a global section that is accessed with GBL\$EDITPSL\$G\_PSL. Metdata can be read from a problem metdata file (thus replacing the OBSERV.MET file) in the directory specified by MAKEPMF\$A\_PROBLEM\_METDATA\_ARCHIVE. These logicals do not have to set if observing locations and metdata are read from a STNLOC.MET and an OBSERV.MET file (see MEDIC.NML description for the appropriate control variables). The central longitude and hemispheric indicator

for transforming UTM map projection information (i.e., central longitude and hemisphere indicator) is currently found in the questionnaire that is accessed with the two logicals: ENTQUESTSF\_QUESTIONNAIRE\_CHANGE\_JOURNAL and ENTQUESTSF\_QUESTIONNAIRE\_ITEM. These two logicals do not need to be set if MERIDIAN and HEMI\_NS are set in MEDIC\_CONTROL\_NML.

### PLCNT Logicals

A number of logicals are used by PLCNT to complete the descriptive information for the products and to handle the distribution of products in the ARAC system. This descriptive information relies on the logicals ARAC\$USER\_NAME to determine the account used to generate the product and the SITE\_ID, PROBLEM\_ID and NODE\_ID to determine the execution environment. Both site system and site workstation model products are placed in the directory specified by MODEL\_PROD\$A\_PRODUCTS. The product attributes for the site system products are written to MODEL\_PROD\$F\_ATTRIBUTES\_FILE and the geography attributes are written to MODEL\_PROD\$F\_GEOGRAPHY\_ATTRIBUTES. The product description records for the site workstation products are written to MODEL\_PROD\$F\_SWS\_DESCRIPTIONS. The frame count (to ensure, when combined with the generation date in the product filename, that each product name is unique over the entire operational ARAC system) for the site system products is maintained in MODEL\_PROD\$F\_FRAME\_COUNT and for the site workstation in MODEL\_PROD\$F\_SWS\_FRAME\_COUNT.

The complete association of directories and files with logical names is listed as follows:

TOPOG.NML	MODEL\$A_GRID	
TOPOG*.CNML	MODEL\$A_GRID	
TOPOG*.CGMB	MODEL\$A_GRID	
TOPOG*.GRID	MODEL\$A_GRID	MODEL\$F_GRID
TOPOG.LOG	MODEL\$A_GRID	
regional elevations	TOPO\$A_REGIONAL	TOPO\$F_REGIONAL
MEDIC.NML	MODEL\$A_WINDS	
MEDIC*.CNML	MODEL\$A_WINDS	
STNLOC.MET	MODEL\$A_WINDS	
STNLOC*.CMET	MODEL\$A_WINDS	
problem stations		EDITPSL\$F_PROBLEM_STATION_LIBRARY
OBSERV.MET	MODEL\$A_WINDS	
OBSERV*.CMET	MODEL\$A_WINDS	
problem metadata	MAKEPMF\$A_PROBLEM_METADATA_ARCHIVE	
GWC grids	DECAFGWC\$A_AFGWC_GRID_FILE	
MEDIC*.CGMB	MODEL\$A_WINDS	

MEDVEL*.BIF	MODEL\$A_WINDS	
MEDIC.LOG	MODEL\$A_WINDS	
MEDIC.VLST	MODEL\$A_WINDS	
questionnaire		ENTQUEST\$F_QUESTIONNAIRE_ITEM
CG-MATHEW.NML	MODEL\$A_WINDS	
CG-MATHEW*.CNML	MODEL\$A_WINDS	
CG-MATHEW*.CGMB	MODEL\$A_WINDS	
MATVEL*.BIF	MODEL\$A_WINDS	
CG-MATHEW.LOG	MODEL\$A_WINDS	
CG-MATHEW.VLST	MODEL\$A_WINDS	
ADPIC.NML	MODEL\$A_DISPERSION	
ADPIC*.CNML	MODEL\$A_DISPERSION	
ADPIC*.RST	MODEL\$A_DISPERSION	
ADPIC*.CGMB	MODEL\$A_DISPERSION	
CONC*.BIF	MODEL\$A_DISPERSION	
CONC*.VLST	MODEL\$A_DISPERSION	
CONC*.RST	MODEL\$A_DISPERSION	
PARTPOS*.BIF	MODEL\$A_DISPERSION	
ADPIC.LOG	MODEL\$A_DISPERSION	
ADPIC.CLST	MODEL\$A_DISPERSION	
MEDIC.VLST	MODEL\$A_VLST	
CG-MATHEW.VLST	MODEL\$A_VLST	
CONC_ANALYSIS.NML	MODEL\$A_DISPERSION	
PLCNT*.CNML	MODEL\$A_DISPERSION	
PLCNT*.CGMB	MODEL\$A_DISPERSION	
PLCNT*.CLST	MODEL\$A_DISPERSION	
PLCNT_*.TBB	MODEL\$A_DISPERSION	
PLCNT.LOG	MODEL\$A_DISPERSION	
user name		ARAC\$USER_NAME
SS products	MODEL_PROD\$A_PRODUCTS	
SWS products	MODEL_PROD\$A_PRODUCTS	
SS attribute records		MODEL_PROD\$F_ATTRIBUTES_FILE
SS geog attributes		MODEL_PROD\$F_GEOGRAPHY_ATTRIBUTES
SWS product descriptions		MODEL_PROD\$F_SWS_DESCRIPTIONS
SS frame count		MODEL_PROD\$F_FRAME_COUNT
SWS frame count		MODEL_PROD\$F_SWS_FRAME_COUNT.
(SS = Site System, SWS = Site Workstation System)		



## APPENDIX E. ARAC Model Namelist Summary

The following tables contain alphabetically sorted lists of all namelist parameters for each model, including their dimensions and input types (real, integer, character). The MEDIC table also indicates the namelist variables that are used in hemispheric as well as regional MEDIC. Array dimensions are given in terms of parameters which are listed in the first table which along with their default values.

<i>DIMENSION PARAMETERS</i>	<i>DEFAULT VALUE</i>	<i>FILE</i>	<i>MODEL</i>
EXPL_CLD_DIM	9	adpic_params.con	adpic
IMAXMAT	51	dims_41_41_15.con	modgrid
JMAXMAT	51	dims_41_41_15.con	modgrid
KMAXMAT	15	dims_41_41_15.con	modgrid
KMAX_UPR_NODDS	9	nodds_dims.con	modutil
KMAX_UPR_HEMI	15	hemi_dims.con	modutil
KMAX_UPR_RWM	10	rwm_dims.con	modutil
MAXCONTLEVELS	5	conc_analysis_dims.con	plcnt
MAXDBINS	10	conc_analysis_dims.con	plcnt
MAX_EXPL_CLD_PROF	30	adpic_params.con	plcnt
MAXFACTORS	15	timehis_dims.con	timehis
MAXFBINS	30	conc_analysis_dims.con	plcnt
MAXHANGLES	10	timehis_dims.con	timehis
MAXLEVEL_ALL	35	upperair_dims.con	medic
MAXPLOTSERIES	5	conc_analysis_dims.con	plcnt
MAXPROF	10	profile_dims.con	modutil
MAX_RESTART	10	adpic_dims.con	adpic
MAXSAMPLERS	600	timehis_dims.con	timehis
MAXSBINS	10	adpic_dims.con	adpic
MAXSOURC	9	adpic_dims.con	adpic
MAXTBINS	5	timehis_dim.con	timehis
MAXTIM	50	adpic_params.con	adpic
MAX_TIMES	10000	CG-MATHEW_specific_params.con	mathew
MAXTIMESTEPS	200	timehis_dims.con	timehis
MAX_VEL	10000	vel_dims.con	modutil
NANGLESTEPS	11	timehis_dims.con	timehis

<i>NAMelist VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
<b>TOPOG_CONTROL:</b>		
CONTOUR_INTERVAL	real	1
FLAT_TOPO	char*1	1
INTERACTIVE	char*1	1
LISTDETAILS	char*1	1
RESTART	char*1	1
<b>TOPOG-PARAMS:</b>		
DELX	real	1
DELY	real	1
DELZ	real	1
DOUBLECELL	char*1	1
MAP_ADJUSTMENT	char*1	1
RANGE	real	1
X_GRIDCENTER	real	1
X_OFFSET_MAT	real	1
X_ORIGIN_PIC	real	1
XRANGE	real	1
Y_GRID_CENTER	real	1
Y_OFFSET_MAT	real	1
Y_ORIGIN_PIC	real	1
YRANGE	real	1
ZRANGE	real	1

<i>NAMelist VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>	<i>SPECIAL</i>
<b>MEDIC_CONTROL:</b>			
GWC_BARB_PLOT_FREQ	integer	1	MEDIC/HMEDIC
GWC_BARB_SCALE	real	1	MEDIC/HMEDIC
HORZ_VECTOR_SCALE	real	1	MEDIC/HMEDIC
IPROFILE	integer	MAXPROF	MEDIC/HMEDIC
JPROFILE	integer	MAXPROF	MEDIC/HMEDIC
MEDIC_VL:ST	char*6	1	MEDIC/HMEDIC
PLOT_LEVELS	integer	KMAXMAT	MEDIC/HMEDIC
PLOT_LEVELS_NHE	integer	KMAX_UPR_HEMI	MEDIC/HMEDIC
PLOT_LEVELS_RWM	integer	KMAX_UPR_RWM	
PLOT_LEVELS_SHE	integer	KMAX_UPR_HEMI	MEDIC/HMEDIC
PRINT_WINDFIELDS	char*1	1	MEDIC/HMEDIC
REGMET_PARAMFILE	char*22	1	
RWM_BARB_PLOT_FREQ	integer	1	MEDIC/HMEDIC
RWM_BARB_SCALE	real	1	MEDIC/HMEDIC
SFC_BARB_PLOT_FREQ	integer	1	
SFC_BARB_SCALE	real	1	
SFC_ROUGH_HGT	real	1	
STNLOCPSL	char*1	1	

TERRAIN_ADJUSTMENT	char*1	1	MEDIC/HMEDIC
VECTOR_PLOT_FREQ	integer	1	MEDIC/HMEDIC
XPROFILE	real	MAXPROF	MEDIC/HMEDIC
YPROFILE	real	MAXPROF	MEDIC/HMEDIC
<i>NAMELIST VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>	<i>SPECIAL</i>

**MEDIC\_PARAMS:**

ALL_WINDS_DIR	real	MAXLEVEL_ALL	MEDIC/HMEDIC
ALL_WINDS_HGT	real	MAXLEVEL_ALL	MEDIC/HMEDIC
ALL_WINDS_SPD	real	MAXLEVEL_ALL	MEDIC/HMEDIC
AVGTIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
BL_HGT	real	1	
EXTRAP_EXPNT_BL	real	1	
MAX_VEER_PARAM_VERT	integer	1	
MAX_VEER_PROF_HORZ	integer	1	
METDATASET	char*40	1	
MET_START_DATE	char*(LEN_DATE_STRING)	1	MEDIC/HMEDIC
MET_START_TIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
NHEDATE	char*(LEN_DATE_STRING)	1	MEDIC/HMEDIC
NHEFCST	integer	1	MEDIC/HMEDIC
NHETIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
PROBMET_DST	char*9	1	
PROBMET_DST_SFC	char*9	1	
PROBMET_DST_TOW	char*9	1	
PROBMET_DST_UPR	char*9	1	
PROFILE_EXTRAP	char*1	1	
PWR_LAW_EXPNT_SL	real	1	
REF_HGT	real	1	
RWMDATE	char*(LEN_DATE_STRING)	1	
RWMFCST	integer	1	
RWMTIME	char*(LEN_TIME_STRING)	1	
SHEDATE	char*(LEN_DATE_STRING)	1	MEDIC/HMEDIC
SHEFCST	integer	1	MEDIC/HMEDIC
SHETIME	char*(LEN_TIME_STRING)	1	MEDIC/HMEDIC
SINGLE_PROFILE_WINDS	char*8	1	
SL_HGT	real	1	
USE_UPR_AT_SFC	char*1	1	

<i>NAMelist VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
<b>CG-MATHEW_CONTROL</b>		
HORZ_VECTOR_SCALE	real	1
IPROFILE	integer	MAXPROF
ISLICE	integer	IMAXMAT
ITER_MESSAGE	integer	
JPROFILE	integer	MAXPROF
JSlice	integer	JMAXMAT
OUTPUT_DESTINATION_DIR	char*10	1
PLOT_LEVELS_ABOVE_TOPO	integer	KMAXMAT
PLOT_GRID_LEVELS	integer	KMAXMAT
PRINT_ADJUSTWIND_ADPIC	char*1	1
PRINT_ADJUSTWIND_DIVERG	char*1	1
PRINT_ADJUSTWIND_FACE	char*1	1
PRINT_ADJUSTWIND_GRID	char*1	1
PRINT_CONVERGENCE	char	1
PRINT_ICCG_CHECK	char*1	1
PRINT_INITIALWIND_DIVERG	char*1	1
PRINT_INITIALWIND_FACE	char*1	1
PRINT_INITIALWIND_GRID	char*1	1
PRINT_LAMBDA_S	char*1	1
PRINT_SIDES	char*1	1
VECTOR_PLOT_FREQ	integer	1
VLST_OPTION	char*6	1
VERT_VECTOR_SCALE	real	1
XPROFILE	real	MAXPROF
XSLICE	real	IMAXMAT
YPROFILE	real	MAXPROF
YSLICE	real	JMAXMAT
<b>CG-MATHEW_PARAMS:</b>		
CONV_TOLERANCE	real	1
ELLIPTIC_EQN_TEST	char*1	1
FIX_LATERAL_INFLOW	char*1	1
FIX_XMAX_INFLOW	char*1	1
FIX_XMIN_INFLOW	char*1	1
FIX_YMAX_INFLOW	char*1	1
FIX_YMIN_INFLOW	char*1	1
ITER_HALT	integer	1
LATERAL_BOUNDARY	char*15	1
MEDVEL_NAMES	char*(TBB_STRING_LEN)	MAX_VEL
MIXED_LATERAL_ADJUST	real	1
MIXED_TOP_ADJUST	real	1
MIXED_XMAX_ADJUST	real	1

MIXED_XMIN_ADJUST	real	1
MIXED_YMAX_ADJUST	real	1
MIXED_YMIN_ADJUST	real	1
PARAMS_DATE	char*(LEN_TIME_STRING)	MAX_TIMES
<i>NAMelist VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
PARAMS_TIME	char*(LEN_DATE_STRING)	MAX_TIMES
SIGMA_HORZ	real	1
SIGMA_VERT	real	MAX_TIMES
STAB_CG-MATHEW	integer	MAX_TIMES
TOP_BOUNDARY	char*15	1
XMAX_BOUNDARY	char*15	1
XMIN_BOUNDARY	char*15	1
YMAX_BOUNDARY	char*15	1
YMIN_BOUNDARY	char*15	1

<i>NAMelist VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
<b>ADPIC_CONTROL:</b>		
DIFF_METHOD	char*5	1
INITIAL_TIME_STEP	real	1
NESTS	integer	1
PARTPOS	char*1	1
PLOT_INTERVAL	char*(LEN_DUR_STRING)	1
PRINT_ADVECT_VEL	char*1	1
PRINT_CELL_CONC	char*1	1
PRINT_FRCTN_VEL	char*1	1
PRINT_MC_DIAG	char*1	1
PRINT_PART_COORD	char*1	1
PRINT_PART_VEL	char*1	1
PRINT_SAMPLING_BINS	char*1	1
PRINT_TOPOG	char*1	1
PRINT_TURB_PARAM_GRADI	char*1	1
PROBLEM_TITLE	char*70	1
RESTART_DATE	char*(LEN_DATE_STRING)	1
RESTART_TIME	char*(LEN_TIME_STRING)	1
RESTART_WRITE_DATE	char*(LEN_DATE_STRING)	MAX_RESTART
RESTART_WRITE_INTERVAL	char*(LEN_DUR_STRING)	1
RESTART_WRITE_TIME	char*(LEN_TIME_STRING)	MAX_RESTART
REFLECT_AT_MLHGT	char*1	1
RUN_DURATION	char*(LEN_DUR_STRING)	1
RUN_START_DATE	char*(LEN_DATE_STRING)	1
RUN_START_TIME	char*(LEN_TIME_STRING)	1
RUN_STOP_DATE	char*(LEN_DATE_STRING)	1
RUN_STOP_TIME	char*(LEN_TIME_STRING)	1
SAMP_CELL_DELZ_FRACT	real	1
SAMPLING_HGT	real	MAXSBINS
SAMPLING_INTERVAL	char*(LEN_DUR_STRING)	1
SAMPLING_TYPE	char*6	MAXSBINS
SKIP_RANDOM_COUNT	integer	1
SOURCES_TO_SAMPLING_BIN	integer	MAXSOURC,MAXSBINS
STATUS_INTERVAL	char*(LEN_DUR_STRING)	1
USE_EXPL_CLD_RISE	char*1	1
USE_VLST	char*1	1
WIND_TYPE	char*6	1
X_NEST_LOC	real	1
Y_NEST_LOC	real	1
X_WINDOW_MIN	real	1
X_WINDOW_MAX	real	1
Y_WINDOW_MIN	real	1
Y_WINDOW_MAX	real	1

Z_WINDOW_MIN	real	1
Z_WINDOW_MAX	real	1

<i>NAMELIST VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
---------------------------	-------------	-------------------

**ADPIC\_METPARAMS:**

DRIFT_VEL_METHOD	integer	1
FSIGU	real	1
FSIGW	real	1
FTLU	real	1
FTLV	real	1
FTLW	real	1
GAUSS_DIFF_MULT_X	real	1
GAUSS_DIFF_MULT_Z	real	MAXTIM
INV_OBUKHOV_LEN_SL	real	MAXTIM
INV_OBUKHOV_LEN_ML	real	1
MAX_DELZ_SHIFT	integer	1
MAX_DITHER	real	1
MET_START_DATE	char*(LEN_DATE_STRING)	MAXTIM
MET_START_TIME	char*(LEN_TIME_STRING)	MAXTIM
MIN_MASS_FRACT	real	MAXTIM
ML_HGT	real	1
PWR_LAW_EXPNT_SL	real	MAXTIM
PRECIP_SCAVENG_COEF	real	MAXTIM
PRECSCAV	real	1
RAIN_START	char*(LEN_DUR_STRING)	1
RAIN_STOP	char*(LEN_DUR_STRING)	1
SFC_ROUGH_HGT	real	1
SIGPHI	real	1
SIGMA_THETA_FACT	real	MAXTIM
STAB_CLASS	integer	MAXTIM
TIMESTEP_FRACT	real	1
TIMESTEP_FRACT_MC	real	1
TURB_PARAMS_TYPE	integer	MAXTIM
INTERP_VEL_WINDS	GORDON	1
U_CALM	real	1

**ADPIC\_SOURCE:**

AMBIENT_TEMP	real	MAXTIM,MAXSOURC
CENTER_HGT	real	MAXTIM,MAXSOURC
EXIT_VEL	real	MAXSOURC
HALFLIFE	real	MAXSOURC
HEAT_EMISSION_RATE	real	MAXSOURC
INV_HGT_AT_STACK	real	MAXTIM,MAXSOURC
PART_DIAM_MAX	real	MAXSOURC
PART_DIAM_MEDIAN	real	MAXSOURC
PART_DIAM_MIN	real	MAXSOURC

PART_DIAM_SGD	real	MAXSOURC
PART_DENSITY	real	MAXSOURC
RELEASE_START	char*(LEN_DUR_STRING)	MAXSOURC
RELEASE_STOP	char*(LEN_DUR_STRING)	MAXSOURC
SFC_DEP_HGT	real	1
SFC_DEP_VEL	real	MAXSOURC
<i>NAMELIST VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
SIGX	real	MAXSOURC
SIGY	real	MAXSOURC
SIGZ	real	MAXSOURC
SOURCE_RATE	real	MAXTIM,MAXSOURC
SOURCE_START_DATE	char*(LEN_DATE_STRING)	MAXTIM
SOURCE_START_TIME	char*(LEN_TIME_STRING)	MAXTIM
STACK_RADIUS	real	MAXSOURC
STACK_TEMP	real	MAXSOURC
TOTPART	integer	MAXSOURC
USE_PLUME_RISE	char*1	MAXSOURC
VERT_TEMP_GRAD	real	MAXTIM,MAXSOURC
XLOC	real	MAXSOURC
XNEG_CUTOFF	real	MAXSOURC
XPOS_CUTOFF	real	MAXSOURC
YLOC	real	MAXSOURC
YNEG_CUTOFF	real	MAXSOURC
YPOS_CUTOFF	real	MAXSOURC
ZNEG_CUTOFF	real	MAXSOURC
ZPOS_CUTOFF	real	MAXSOURC
<b>SIGMA_THETA_DATA:</b>		
SIGMA_THETA	real	MAXTIM, MAXLEVEL_ALL
SIGMA_THETA_HGT	real	MAXTIM, MAXLEVEL_ALL
<b>ADPIC_EXPL_CLD:</b>		
DIF_IN_EXPL_CLD	char*1	1
DTDZ1	real	1
DTDZ2	real	1
HEAMT	real	1
HEATDET	real	1
PLOT_EXPL_CLD	integer	1
PRINT_EXPL_CLD	char*1	1
PRINT_TYPE_EXPL_CLD	char*5	1
REL_HUM	real	1
SIG_OVER_R	real	1
SRC_EXPL_CLD	integer	MAXSOURC,EXPL_CLD_DIM
SRC_PRESS	real	1
TA0	real	1
TEMP_PROFILE	real	MAX_EXPL_CLD_PROF
TEMP_PROFILE_HGT	real	MAX_EXPL_CLD_PROF



UAEXP	real	1
UAREF	real	1
UAZREF	real	1
VEL_COUPL	real	1
ZI	real	1
ZINIT	real	1

<i>NAMELIST VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
<b>CONC_ANALYSIS_CONTROL:</b>		
ANGLE_WIDTH_FACTOR	real	1
CENTERLINE_SOURCE	integer	1
CONC_RESTART_DATE	char*(LEN_DATE_STRING)	1
CONC_RESTART_TIME	char*(LEN_TIME_STRING)	1
DCON	real	MAXFBINS
DOSE_TO_FINAL_BIN	integer	MAXDBINS,MAXFBINS
FULL_SEARCH_FACTOR	real	1
INTEGRATION_STEP	char*(LEN_DUR_STRING)	MAXPLOTSERIES,MAXFBINS
INTEGRATION_TYPE	char*4	MAXFBINS
PLOT_END_DATE	char*(LEN_DATE_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_END_TIME	char*(LEN_TIME_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_START_DATE	char*(LEN_DATE_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_START_TIME	char*(LEN_TIME_STRING)	MAXPLOTSERIES,MAXFBINS
PLOT_STEP	char*(LEN_DUR_STRING)	MAXPLOTSERIES,MAXFBINS
PRINT_CONTOUR_CONC	char*1	1
PRINT_INPUT_CONC	char*1	1
RADIAL_STEP_FACTOR	real	1
RESUSPEND	char*1	1
SAMPLING_TO_DOSE_BIN	integer	MAXDBINS
SECTOR_ANGLE_MAX	real	1
SECTOR_ANGLE_MIN	real	1
<b>PLCNT_GRAPHICS_PARAMS:</b>		
BLOWUP_SOURCE	integer	1
CIRCLE_2000	char*1	1
CONTOUR_LEVELS	real	MAXCONTLEVELS,MAXFBINS
HALF_POWER_OF_10	char*1	1
INITIAL_BANNER	char*1	1
MAKE_NEW_GEOG	char*1	1
MAPSCALE	integer	2
MAP_SCALE_FLAG	char*1	MAXFBINS
MAX_CONTOURS	integer	MAXFBINS
MIN_CONTOUR_AREA	real	1
NEST_CONTOURS	char*1	1
NEST_TYPE	char*10	1

NOTES_BANNERS	char*1	1
NUM_RECOMPUTED_CONTOURS	integer	1
OUTPUT_CONTOURS	char*1	1
OUTPUT_GRIDS	char*1	1
RECOMPUTE_USER_CONTOURS	char*1	1
SOURCE_OCTANT	char*2	1
TITLE_ON_GEOG	char*1	1
XMAX_WINDOW	real	1
XMIN_WINDOW	rea	1
XY_TICK_LABEL	char*1	1
YMAX_WINDOW	real	1
YMIN_WINDOW	real	1

<i>NAMELIST VARIABLES</i>	<i>TYPE</i>	<i>DIMENSIONS</i>
---------------------------	-------------	-------------------

**PLCNT\_LABELING\_PARAMS:**

DESCR_FORMAT	char*5	MAXFBINS
DESCR_HEADERA	char*32	MAXFBINS
DESCR_HEADERB	char*32	MAXFBINS
GENERATION_DATE	char*(LEN_DATE_STRING)	1
GENERATION_TIME	char*(LEN_TIME_STRING)	1
LEGEND_TITLE	char*32	MAXFBINS
LEGEND_TITLE_ALL	char*32	1
LEVEL_DESCRit	char*32	5*MAXFBINS
MATERIAL	char*22	MAXFBINS
PROBLEM_TITLE_PLCNT	char*70	1
PLOT_DESCR	char*32	MAXFBINS
PLOT_TITLE	char*32	MAXFBINS
REMARKSt	char*32	4*MAXFBINS
REMARKSt_ALL	char*32	4
SITESYS_LOCATION_NUM	integer	1
SOURCE_LOCATION_NUM	integer	1
SOURCE_LOCATION_TYPE	char*6	1
UNITS	char*10	MAXFBINS

**TIMEHIS\_PARAMS:**

BACKGROUND_CALC	real	MAXFBINS
BACKGROUND_CALC_ALL	real	1
BACKGROUND_MEAS	real	MAXFBINS
BACKGROUND_MEAS_ALL	real	1
BIN_PLOT_MAX	real	MAXFBINS
BIN_PLOT_MAX_ALL	real	1
BIN_PLOT_MIN	real	MAXFBINS
BIN_PLOT_MIN_ALL	real	1
COMPARE_MEAS_CALC	char*1	1
FACTOR_R	real	MAXFACTORS
HIT_ANGLE	real	MAXHANGLES

MIN_CONC	real	MAXFBINS
MIN_CONC_ALL	real	1
PROBLEM_TITLE_TIMEHIS	char*70	1
SHOW_TIME_ORIGIN	char*1	1
TIMESTEPS_PER_PLOT	integer	1
XMAX_SAMPLES	real	1
XMIN_SAMPLES	real	1
YMAX_SAMPLES	real	1
YMIN_SAMPLES	real	1